Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 370 – 383
© TÜBİTAK
doi:10.3906/elk-1304-123

A sparsity-preserving spectral preconditioner for power flow analysis

Emrullah Fatih YETKİN[1,*], **Hasan DAĞ**[2]
[1]Caferağa Mah., Dalga Sok., Eren Apt. No 4 D 1, Kadıköy, İstanbul, Turkey
[2]Department of Management Information Systems, Faculty of Engineering and Natural Sciences,
Kadir Has University, İstanbul, Turkey

Received: 14.04.2013 • **Accepted/Published Online:** 04.10.2013 • **Final Version:** 05.02.2016

Abstract: Due to the ever-increasing demand for more detailed and accurate power system simulations, the dimensions of mathematical models increase. Although the traditional direct linear equation solvers based on LU factorization are robust, they have limited scalability on the parallel platforms. On the other hand, simulations of the power system events need to be performed at a reasonable time to assess the results of the unwanted events and to take the necessary remedial actions. Hence, to obtain faster solutions for more detailed models, parallel platforms should be used. To this end, direct solvers can be replaced by Krylov subspace methods (conjugate gradient, generalized minimal residuals, etc.). Krylov subspace methods need some accelerators to achieve competitive performance. In this article, a new preconditioner is proposed for Krylov subspace-based iterative methods. The proposed preconditioner is based on the spectral projectors. It is known that the computational complexity of the spectral projectors is quite high. Therefore, we also suggest a new approximate computation technique for spectral projectors as appropriate eigenvalue-based accelerators for efficient computation of power flow problems. The convergence characteristics and sparsity structure of the preconditioners are compared to the well-known black-box preconditioners, such as incomplete LU, and the results are presented.

Key words: Iterative methods, power flow analysis, spectral projectors, Krylov accelerators, sparse approximation

1. Introduction

Obtaining the bus voltages (both magnitudes and phase angles), branch currents, and both the active and reactive power flows in all branches of a power system under given conditions is called the power flow problem [1]. This problem is very important for both short- and long-term planning and operational purposes of the power system at hand. The bus voltages and line admittances can be expressed in polar forms, and one can obtain a set of nonlinear equations for the active and reactive powers [1]. The Newton–Raphson (NR) method is one of the most popular methods for solving these types of nonlinear equations [2]. At each NR iteration step, the Jacobian matrix has to be evaluated, and a linear set of equations, whose coefficient matrix is the Jacobian matrix, has to be solved. The general form of the linear equation set appearing in each NR step is given below.

$$\begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \tag{1}$$

These equations are in the form of Ax = b and they represent a step of the power flow problem. The notation Ax = b will be employed to show the NR steps in this paper. One can find a wide set of applications and solution

[*] Correspondence: efatihyetkin@gmail.com

methods regarding power flow analysis in the literature [3–5]. The linear equation set given in Eq. (1) is traditionally solved with the sparse direct methods [2]. Whereas direct solvers are robust methods [6], iterative methods have also been shown to be almost as robust as direct solvers if a suitable preconditioner is used. In this paper, we mainly focus on the development of a preconditioner for the generalized minimum residual (GMRES) method because of the nonsymmetric structure of the Jacobian matrices used. For large and sparse systems, Krylov subspace methods (conjugate gradient, GMRES, etc.) have advantages over direct methods in terms of better parallelism and less memory requirement [7,8]. However, in most cases, a preconditioner is essential for accelerating the convergence rate of Krylov subspace-based iterative methods [9,10].

In the proposed method, the extreme eigenvalues of the Jacobian matrix are removed with a spectral preconditioner. The proposed preconditioner is mainly based on the idea given in [11,12], yet it is computationally more effective and efficient. The method has similar needs as the preconditioner given in [12], such as rough data about the distribution of the eigenvalues of matrix A. This information can easily be obtained with the eigenvalue inclusion theorems, such as the Gershgorin theorem or the well-known power iteration [13]. With this approximate information about the spectral distribution of matrix A, one can define a curve integral to obtain a spectral projector, which is an invariant subspace corresponding to the extreme eigenvalues of the Jacobian matrix. The most time-consuming part of the suggested preconditioner is the computation of the spectral projector, which requires a matrix inversion. The sparse approximate inverse (SPAI) method can be used to obtain an approximate spectral projector to create the proposed preconditioner. The SPAI technique is an efficient preconditioner implementation for sparse linear systems [14]. In the power flow problem, the eigenvalues of the Jacobian matrix in NR steps do not vary significantly [2]. Thus, the preconditioner can be computed in the first step of the NR iteration and can be used in the subsequent NR iterations. Another advantage of the proposed preconditioner over the traditional preconditioners (such as incomplete LU, or ILU) is its sparsity preservation capability. Finally, the building blocks of the approach (such as SPAI, matrix multiplication, or partial SVD) are all well-known matrix operations, and the method itself is easy to parallelize.

In Section 2 of this paper, the method is introduced. Several test results and comparisons with other methods are given in Section 3. Finally, the conclusions are presented in the last section.

2. Sparsity-preserving computation of spectral projectors

2.1. Introduction to spectral projectors

Let $A \in \Re^{nxn}$ be given and its eigenvalue spectrum be composed of 2 separate parts as $\Lambda(A) = \Lambda_1 \cup \Lambda_2$, and let the intersection of these 2 parts be an empty set. One can then define an **A**-invariant subspace **Z**, named as a spectral projector corresponding to Λ_1. The matrix sign function (MSF) is a well-known spectral projector [15]. The MSF behaves like its scalar equivalent. Let the Jordan canonical form of matrix **A** be given as:

$$A = X \begin{bmatrix} J^{+} & 0 \\ 0 & J^{-} \end{bmatrix} X^{-1}, \tag{2}$$

where the eigenvalues of J^{+} are equal to the positive eigenvalues of matrix **A** and the negative eigenvalues of matrix **A** are equal to the eigenvalues of J^{-}. The columns of matrix **X** represent the corresponding eigenvectors. The MSF of matrix **A** can be written similarly to Eq. (2):

$$sign(A) = X \begin{bmatrix} I_m & 0 \\ 0 & I_{n-m} \end{bmatrix} X^{-1}, \tag{3}$$

where m represents the number of the eigenvalues with positive real parts [16]. MSF can also be defined as a contour integral:

$$sign(A) = 2P(A) - I, \tag{4}$$

where **I** is the identity matrix with the appropriate dimensions, and the matrix-valued function **P** is defined as follows:

$$P(A) = \frac{1}{2\pi i} \oint\limits_{\Gamma} (zI - A)^{-1} dz. \tag{5}$$

In Eq. (5), Γ can be any closed curve that contains the desired eigenvalues of matrix **A** [17]. For MSF, Γ needs to enclose all the eigenvalues whose real parts are negative [18]. MSF is an efficient and stable tool for the spectral dichotomy of a given matrix [19]. On the other hand, the computational cost of MSF can be very high due to the matrix inversions in its computational schemes [20–23]. In their previous work, the authors proposed a fast way for computing an approximate MSF so as to use it as a preconditioner [12]. The method proposed in [12] is essentially based on replacing the direct inversion given in Eq. (6) with SPAI.

$$A_{k+1} = \frac{1}{2}(A_k + A_k^{-1}), A_0 = A \tag{6}$$

The iteration given in Eq. (6) is called a Newton iteration and it is quadratically convergent [16]. Due to the usage of SPAI in the algorithm, the convergence rate will decrease without any a priori prediction [12].

In this work, approximate and sparsity-preserving spectral projectors will be computed by the explicit integration of the contour integral given in Eq. (5). To handle the computational complexity of the problem, SPAI and some drop tolerance-based approaches are employed.

2.2. Computation of spectral projectors

In Eq. (5), Γcan be any preselected geometrical domain. The most appropriate integration method for Eq. (5) is the Gauss quadrature [24]. Let us use a polygon contour with m edges as Γ. Then the integral can be written for each edge with appropriate boundaries and with changes of variables as:

$$P(A) = \frac{1}{2\pi i} \sum_{j=1}^{m} \int_{z_j}^{z_{j+1}} (zI - A)^{-1} dz. \tag{7}$$

After the evaluation of this integral with Gaussian quadrature in a few points, the spectral projector can be used to decompose the matrix spectrum [25]. The number of eigenvalues of matrix **A** in the preselected contour Γ can be computed by the trace of the projector **P** [25].

In Eq. (7), one can apply the change-of-variables method to define the edges. To realize this for the rectangular domain given in Figure 1, domain Γ is divided into 4 subdomains. For each subdomain, variable z changes to variable t to specify the edges. These transformations are given below:

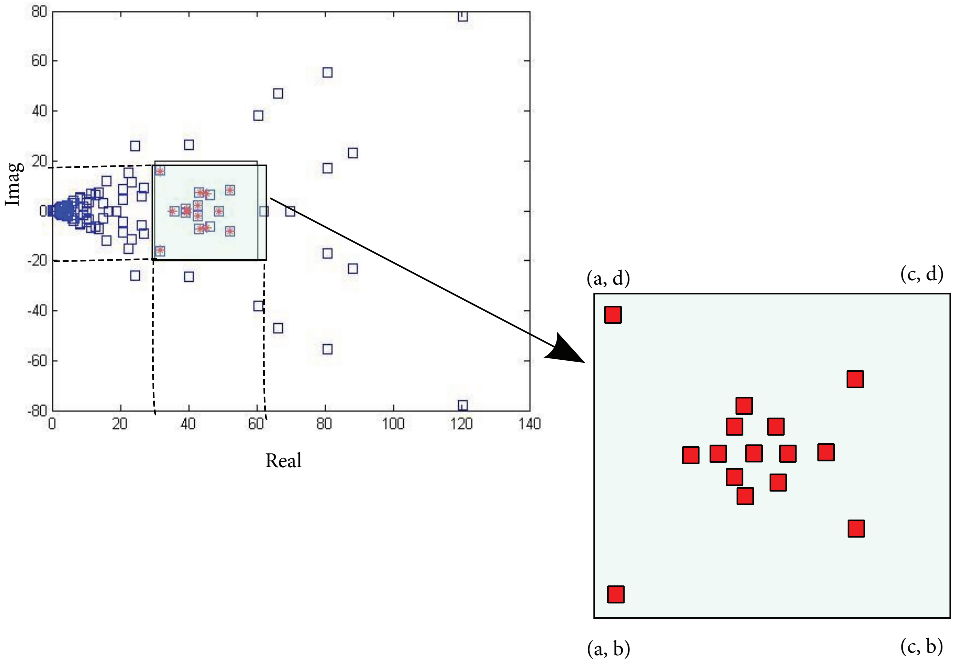

Figure 1. Edge points of the enclosure when Γ is selected as a rectangular.

$$\begin{aligned} \Gamma_1 &: z = (1-t)[a+ib] + t[c+ib] \\ \Gamma_2 &: z = (1-t)[c+ib] + t[c+id] \\ \Gamma_3 &: z = (1-t)[c+id] + t[a+id] \\ \Gamma_4 &: z = (1-t)[a+id] + t[a+ib] \end{aligned}, \tag{8}$$

where $t \in [0,1]$. Obviously, the same transformation needs to be applied to the derivational part of the integration. Thus, in each edge, dz will be replaced by:

$$\begin{aligned} \Gamma_1 &: dz = (c-a)dt \\ \Gamma_2 &: dz = i(d-b)dt \\ \Gamma_3 &: dz = (a-c)dt \\ \Gamma_4 &: dz = i(b-d)dt \end{aligned}. \tag{9}$$

If all these changes are substituted into Eq. (7), one can obtain the integral below forΓ_1:

$$\int_0^1 (\varphi_1(t)I - A)^{-1}(b-a)dt, \tag{10}$$

where $\varphi_1(t) = (1-t)(a+ic) + t(b+ic)$. All other integrals can be similarly derived. The parameters (a, b, c, d) and the shape of the domain can be seen in Figure 1.

2.3. Efficient computation of spectral projectors with SPAI

Several matrix inversions are needed for the computation of spectral projectors. Fortunately, for most physical systems, the matrices have sparse structures. For example, in power flow or circuit simulation problems in electrical engineering, the system matrices have only a few entries in each column. Hence, it is logical to benefit from the sparse structure of the matrices for spectral projector computation. For this purpose, exact matrix inversions can be replaced with sparse approximate inversions. There are several ways to compute an approximate inverse of a sparse matrix [26]. In this work, we employ the SPAI method, which was developed by Barnard et al. [14]. The SPAI method computes an approximate inverse of a sparse matrix **A,** denoted by **K**, with the same sparsity structure as **A**, by minimizing the Frobenius norm of $||$AK-I$||_F$. The computation of a spectral projector for a sparse matrix **A** with SPAI is listed in Algorithm 1. In the algorithm, a rectangular domain is selected as Γ. The coordinates of the corners of the rectangular domain are shown in Figure 1. The algorithm can be modified easily for circular domains using the identity given in [27].

Algorithm 1. Building the sparsity-preserved projector.

Input: Matrix **A**, coordinates of the corner (a,b,c,d), and number of integration points (k).

Output: Spectral projector **P.**

```
1. Select k integration points from the edges of the rectangular according to Gauss quadrature rules
2. for i = 1:4 (for each edge of the rectangular) do
3. for j = 1:k (for each integration point) do
      a. Prepare the matrix $T_{ij} = (\varphi_i(t_j)I - A)$according to the change of variables given in Eqs. (8) and (9)
      b. Convert the complex matrix$T_{ij}$into the double-size real matrix for SPAI as:
         $$T_{ij}^{'} = \begin{bmatrix} \Re(T_{ij}) & \Im(T_{ij}) \\ -\Im(T_{ij}) & \Re(T_{ij}) \end{bmatrix}$$
      c. Compute the sparse approximate inverse$K_{ij}^{'} = spai(T_{ij}^{'})$and compute$K_{ij} = K_{ij}^{'}(1:n,1:n) + jK_{ij}^{'}(1:n,n+1:2n)$, where $n$ shows the dimension of matrix A, and j equals the square root of –1.
   endfor
4. $P_i = \sum_{j=1}^{k} K_{ij}$
endfor
5. $P = \frac{\sigma}{2\pi j}\sum_{i=1}^{4} P_i$ where $\sigma$ arises from the geometry
```

The proposed algorithm has 2 main limitations. The algorithm needs a double-size matrix inversion; however, this situation is handled by the sparse approximate inversion without a large computational cost. The second limitation is the number of integration points. If the number of integration points is increased, the computational cost will also increase. Therefore, optimal selection of integration scheme is important for the efficiency of the algorithm. After several experimental results, the authors suggest using the Gauss quadrature for the numerical integration. To realize this algorithm, the SPAI code developed by Grote et al. is used, and there are some important parameters in their method. For example, parameter ε controls the quality of the approximation and takes a value between 0 and 1. Smaller ε values usually give more accurate, yet slower, inversions [14]. It is observed that an average value for ε between 0.4 and 0.6 gives satisfactorily accurate results for the computed eigenvalues. In the numerical experiments section, there is a comparison of the different ε values and the accuracy of the computed eigenvalues. Furthermore, the relationship between the accuracy and the number of Gaussian quadrature points is investigated, and our observation is that the desired eigenvalues can be found without dramatically increasing the number of integration points.

2.4. Computation of the eigenvalues in the selected domain with the sparsity-preserving spectral projectors

In most engineering problems, only a small subset of the eigenvalues is sufficient to determine the behavior of the system at hand. For example, the stability of an electrical power system can be determined by the nonexistence of an eigenvalue with a positive real part. Hence, a power system engineer may need a tool for tracking the behaviors of the real parts of some critical eigenvalues of the system matrices according to changes in the system [28].

In this paper, we employ an approximate sparsity-preserving spectral projector to build a preconditioner. The aim of the proposed preconditioner is to vanish the extreme eigenvalues for a reasonable reduction in the condition number of the Jacobian matrix. To realize this, it is necessary to select the appropriate coordinates to compute the spectral projector with Algorithm 1. These coordinates can be estimated via Gershgorin circles and/or with a few steps of power iteration [12]. Then the approximate spectral projector $\mathbf{P}$ can be employed to create the block diagonal form of matrix $\mathbf{A}$, whose diagonal blocks have separate spectral information. The algorithm is listed in Algorithm 2. Two different examples for the IEEE 300-bus test case can be found in Figures 2 and 3.

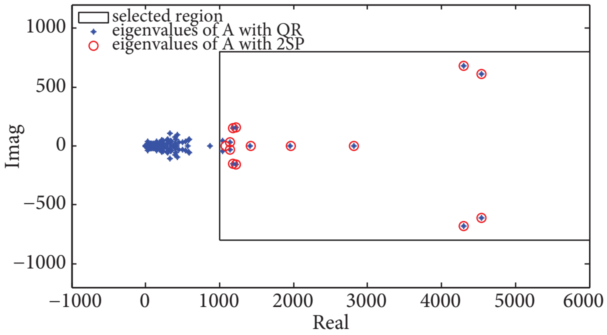

Figure 2. Γ is selected as rectangular with (a = 1000, b = –800, c = 6000, d = 800) and the ε value for SPAI is selected as 0.1. The number of integration points for Gauss quadrature is selected as 10. The method found most of the eigenvalues in Γ. The sparsity index of the matrix **U** is .0.044.

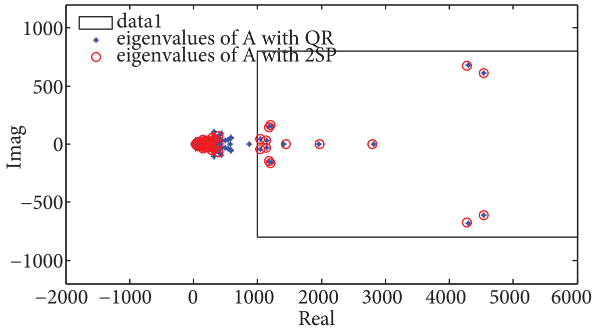

Figure 3. $\Gamma$is selected as rectangular with (a = 1000, b = –800, c = 6000, d = 800) and the ε value for SPAI is selected as 0.7. The number of integration points for Gauss quadrature is selected as 10. The method found most of the eigenvalues in Γ. The sparsity index of the matrix **U** is 0.015.

SVD decomposition is the most time-consuming part of the algorithm. However, only left singular vectors are used to obtain diagonal blocks; hence, one can reduce the computational cost to half. On the other hand, matrix **U**, which consists of the left singular vectors, will be dense. It is possible to use drop tolerance to save the sparsity of the result matrices. If the entries of matrix **U** are smaller than a tolerance, then they are dropped. Consequently, both the computational effort and the number of nonzero elements will be decreased.

```
Algorithm 2. Block decomposition with spectral projectors.
Input: $\mathbf{P}$ (spectral projector).
Output: Block diagonal form of the $\mathbf{A}$.

1. Find the trace of projector $\mathbf{P}$ as $r$. It is equal to the number of eigenvalues in the selected domain
2. Compute the SVD decomposition of the projector $\mathbf{P} = \mathbf{USV}^H$
3. Let $\mathbf{U}_r = \mathbf{U}(:,1:r)$ and $\mathbf{U}_p = \mathbf{U}(:\ , r + 1:n)$
4. Compute the transformations $\mathbf{A}_{11} = \mathrm{U}_r^H \mathrm{AU}_r$ and $\mathbf{A}_{22} = \mathrm{U}_p^H \mathrm{AU}_p$
5. $\Lambda(A_{11}) = \Lambda_1,\ \Lambda(A_{22}) = \Lambda_2$, where$\Lambda_1,\ \Lambda_2$ show the extreme eigenvalues and the rest, respectively
```

2.5. Building the preconditioner

To achieve more robust and rapid convergence for an iterative method, a preconditioner should be used. Preconditioners have been thoroughly studied in the literature [10,29]. Preconditioning mainly aims to decrease the condition number of the coefficient matrix. To do this, the number of eigenvalue groups of matrix **A** should be decreased. In this work, orthogonal spectral projectors are employed to reorder and deflate the outcome of the maximal and minimal eigenvalues on the condition number of the coefficient matrix ***A***. The left, right, and two-sided preconditioned linear systems are given in Eq. (11):

$$\begin{aligned} &M^{-1}Ax = M^{-1}b \\ &(AM^{-1})Mx = b \\ &M_L^{-1}(AM_R^{-1})M_R x = M_L^{-1}b \end{aligned} \quad . \tag{11}$$

In the proposed method, sparsity-preserving spectral projectors (2SP) are used to produce an orthogonal projector for transforming matrix ***A*** into the block form given in Algorithm 2. After that operation, one can define a preconditioner for the iterative method. GMRES is selected as the iterative method in this work. After a suitable (a,b,c,d) coordinate set is selected with the help of either the power iteration or the Gershgorin discs, Algorithm 1 may be used to obtain the spectral projector. In the second step, the block diagonal form of matrix **A** can be obtained using Algorithm 2. The result form of matrix **A** can be given as follows:

$$U^T AU = B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \tag{12}$$

Theoretically, the eigenvalues of matrix $\boldsymbol{B}_{\mathbf{11}}$ are equal to those of ***A*** in the selected domain Γ. When the SPAI method and drop tolerance are used, however, the eigenvalues of $\boldsymbol{B}_{\mathbf{11}}$ are no longer equal to the eigenvalues of **A**, although they are approximately equal, and this is usually acceptable for a preconditioner. Moreover, the norm of the $\boldsymbol{B}_{\mathbf{21}}$ will be approximately equal to zero**.** Hence, we can say that the eigenvalues of $\boldsymbol{B}_{\mathbf{22}}$ will be approximate enough to the rest of the eigenvalues of matrix **A**. More precisely, the preconditioner matrix for the first approach can be given as:

$$M = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix}, \tag{13}$$

where the dimension of $\boldsymbol{B_{11}}$ and $\boldsymbol{B_{22}}$ equals k and *(n – k)* $(n - -k)$, respectively. Here, the eigenvalues of matrix $\boldsymbol{B_{11}}$ are approximately equal to the eigenvalues of matrix $\boldsymbol{A}$ in domain Γ, and $\boldsymbol{B_{22}}$ is an identity matrix with appropriate dimensions. If the dimensions of $\boldsymbol{B_{11}}$ and $\boldsymbol{B_{22}}$ get closer to each other, the computation and implementation of the preconditioner can be performed in parallel with a high degree of efficiency. After the **U** matrix is applied onto $\boldsymbol{Ax} = \boldsymbol{b}$, we get

$$U^T AU(U^T x) = U^T b. \tag{14}$$

In Eq. (14), matrix $\boldsymbol{U}$ is multiplied by $\boldsymbol{A}$ from both sides to ensure that the eigenvalues of $\boldsymbol{A}$ are in the same order as the preconditioner matrix $\boldsymbol{M}$. Finally, the original system is transformed into $\boldsymbol{A_n x_n} = \boldsymbol{b_n}$, where:

$$\begin{array}{l} A_n = U^T AU \\ b_n = U^T b \qquad . \\ x_n = U^T x \end{array} \tag{15}$$

The new linear system can be solved by any suitable iterative method with a preconditioner $\boldsymbol{M}$. The total algorithm is listed in Algorithm 3.

Algorithm 3. Sparsity-preserving spectral preconditioner.

Input: Matrix **A**, right-hand-side vector b, drop tolerance dt.

Output: (i) Preconditioner **M** and (ii) solution of the **A**x = b using the preconditioner **M**.

(i) Obtaining the preconditioner:

1. Employ Gershgorin discs or power iteration to obtain a reasonably closed curve $\Gamma$with its coordinates (a, b, c, d)
2. Use Algorithm 1 to obtain the spectral projector P
3. Compute the left singular space **U** with drop tolerance dt and block diagonal form of matrix **A**, as given in Eq. (12)
4. Build preconditioner matrix **M** as

$$M = \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix}$$

(ii) Solving systems using preconditioner **M**:

1. Compute $\mathbf{A_n}$ and b_n using Eq. (15)
2. Use an iterative method to solve linear equations with preconditioner $\mathbf{M_b}$ and obtain $x_{n.}$
3. Retrieve the true solution x = $\mathbf{Q_b} x_n$

3. Numerical results

3.1. Sparsity preservation and accuracy of 2SP approach

Well-known IEEE power system benchmark examples in MATPOWER [30] are employed to show the efficiency of the proposed method. With the NR method, a new Jacobian matrix will be created in each step. The Jacobian matrix, computed in the first step of NR iteration, is used in the following experiments. As shown in [2], the eigenvalues of the Jacobian matrix do not change dramatically in subsequent NR steps [2]. This is an important advantage of the proposed preconditioner. The preconditioner is created at the very beginning as a preprocessing step, and is used for subsequent linear systems, with the Jacobian matrix being the coefficient matrix. In Table 1, one can find the basic properties of the test matrices used in the experiments.

Table 1. Some numerical properties of test cases.

Number of buses	Matrix size	Number of nonzeros	Cond (J)
118	181	1051	3.17×10^3
300	530	3736	1.16×10^5
2383	4438	27,803	9.69×10^5
2746	5127	32,120	4.59×10^6

Sparsity preservation capability of the proposed method is also significant, and its importance will increase if the dimension of the Jacobian is ascending. To show the preservation capability and accuracy of the proposed method, we tested the algorithm in 2 different cases. To compare the sparsity structures, we define a quantity for sparsity as:

$$sparsity = \frac{nnz(A)}{n^2}. \tag{16}$$

In the first test, the extreme eigenvalues of the 300-bus system are captured and the sparsity structures of matrix $\mathbf{B}$ for 2 different ε selections are investigated.

As a second test, the sparsity structures of ILU preconditioners and the proposed 2SP preconditioner are compared for 300- and 2383-bus systems. ILU methods basically calculate an approximation of the classical LU decomposition, and these incomplete L and $\boldsymbol{U}$ factors are employed as preconditioners [29]. There are several different implementations of ILU factorization that can be found in the literature.

1. *ILU threshold*: The entries below a predefined threshold value of $\boldsymbol{L}$ and $\boldsymbol{U}$ factors are dropped, and the resultant approximate $\mathbf{L}$ and $\mathbf{U}$ factors are used as preconditioners.

2. *ILU('x'):* The sparsity pattern of matrix $\boldsymbol{A}$ determines the dropping strategy of fill-ins. For example, in the ILU(0) method, no fill-in is allowed outside the sparsity pattern of matrix $\mathbf{A}$.

As can be observed from Figures 4 and 5, the sparsity indices of 2SP preconditioners are lower than those of ILU-type preconditioners for each case. Especially for the 2383-bus case, the sparsity index does not change with respect to the number of integration points. The third test result of the sparsity preservation capability of the 2SP method shows the sparsity structure for 2383-bus system preconditioners and the relative residuals for the GMRES iteration. These results are shown in Figure 6. In these tests, the threshold value for ILU is taken as 0.00001 for rapid convergence, and *eps* is taken as 0.1 for accuracy. The restart value for GMRES iteration is 10 and the tolerance is 10^{-8}.

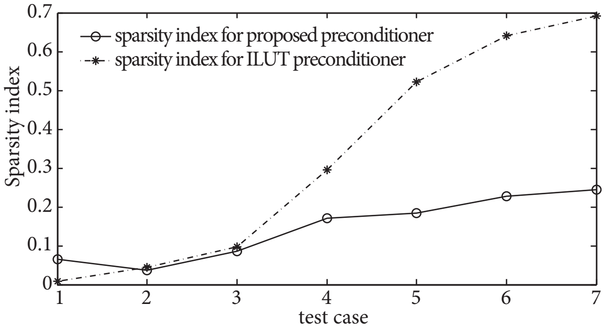

Figure 4. Comparison of the 300-bus system Jacobian, the sparsity indices. They are compared for threshold value for ILU-type preconditioners and number of integration points for 2SP preconditioner, with Γ selected as rectangular with (a = 1000, b = –800, c = 6000, d = 800).

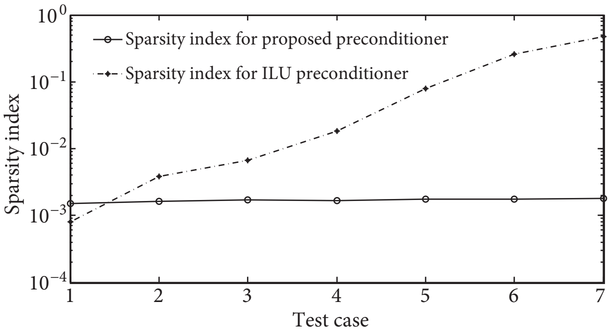

Figure 5. Comparison of the 2383-bus system Jacobian, the sparsity indices. They are compared for threshold value for ILU-type preconditioners and number of integration points for 2SP preconditioner, with selected as a rectangular with (a = 12000, b = –1000, c = 25000, d = 1000).

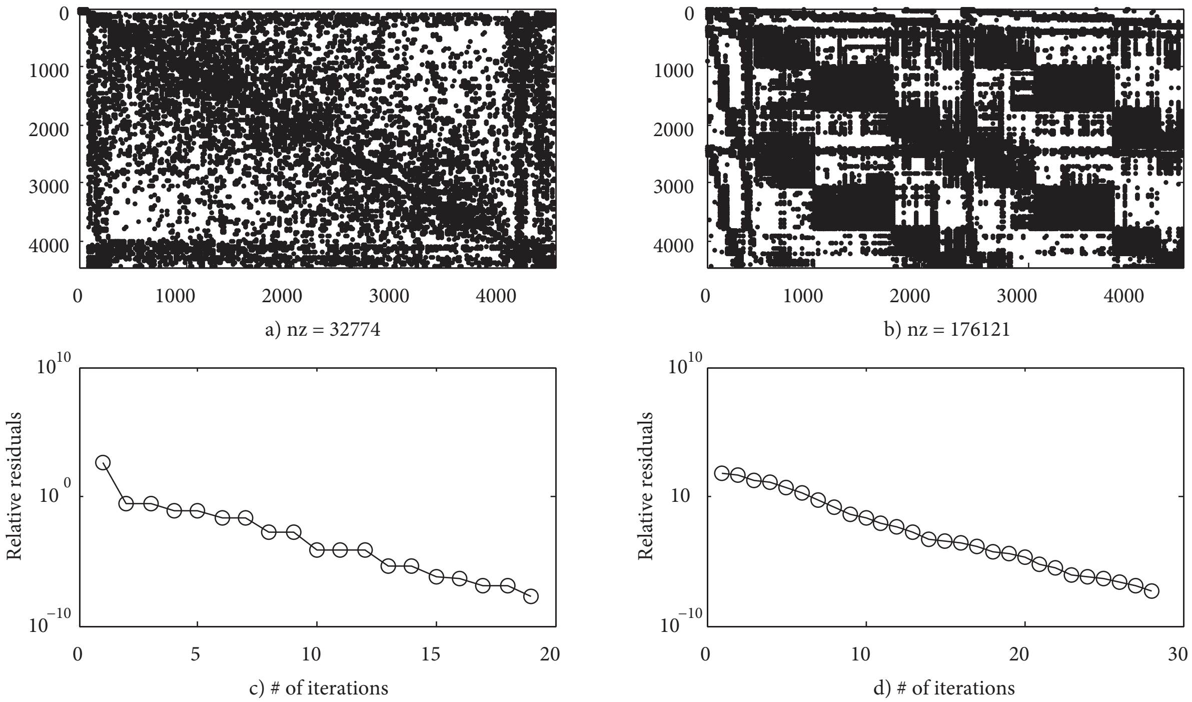

Figure 6. Comparison of the sparsity structures of the preconditioners and residual histories for 2383-bus test case. In (a) and (c), 2SP is used with $\varepsilon = 0.1$ and 10 integration points, In (b) and (d), ILUT is used with 0.0001 drop tolerance.

3.2. Comparison between 2SP and several types of ILU

The residual history of the restarted GMRES algorithm with several types of ILU and 2SP preconditioners is shown in Figure 7. In all tests, the restart value is chosen as 10, stopping tolerance of GMRES is chosen as 10^{-8}, and maximum iteration number is set to the dimension of the Jacobian matrix. All computations are performed on a Pentium 4 processor with 2 GB of RAM.

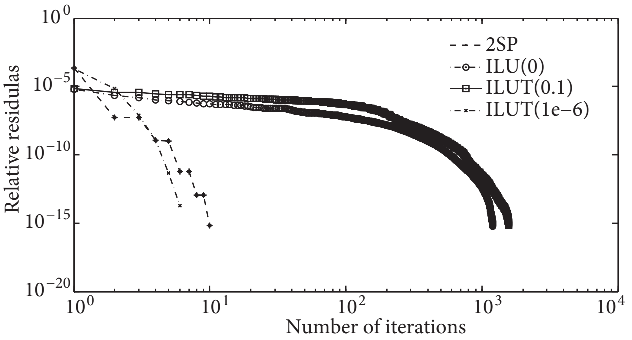

Figure 7. The residual histories for the first Jacobian of the 2746-bus artificial Poland electricity system data with several preconditioners. The convergence behavior of the proposed 2SP preconditioner is very close to the ILUT (0.000001) preconditioner. On the other hand, because of the small drop tolerance, the ILUT preconditioner is very close to the direct solution and its sparsity is lost.

The proposed preconditioners are compared with classical ILU methods in the NR iterations of power flow simulation. To implement this, MATPOWER is employed [30]. The default linear equation solver is the classical LU method in the MATPOWER package. To test the proposed preconditioner, the LU solver is replaced with MATLAB's GMRES package [31]. In our tests, we used the Poland electricity system data, which has 2383 and 2746 buses. Several important properties of these data are given in Table 1.

The preconditioner matrix was created only once and was used in all other NR steps. For IEEE-300, -2383, and -2746 test cases, we observed that GMRES with ILU (0) and ILUT (0.1) did not converge with the solution. As a result, satisfactory accelerations with the proposed 2SP-based preconditioners were obtained.

The effectiveness of the proposed preconditioners can be also observed in Tables 2 and 3. The preconditioner is produced in the first step of the NR iteration and is used in the other iterations. In the literature, it is proved that the eigenvalues of the Jacobian matrix do not change widely for power flow analysis problems. Therefore, 2SP preconditioners can be computed only once, and then the same preconditioner can be used in each step of the NR iteration.

Table 2. Iteration number in each NR step with the same preconditioner for every step. The test case is the Poland electricity system model with 2383 buses.

NR-iteration	ILU (0)		ILUT (0.001)		ILUT (10^{-6})		2SP	
	Outer	Inner	Ouvter	Inner	Outer	Inner	Outer	Inner
1	34	4	3	4	1	4	2	6
2	44	3	5	3	1	7	2	6
3	35	5	9	2	1	5	2	2
4	36	1	8	5	1	5	2	4
5	36	4	8	3	1	5	2	4
6	44	10	7	5	1	5	2	5

4. Conclusion

A new preconditioner design based on sparsity-preserving spectral projectors is proposed in this study. The proposed 2SP algorithm is tested and compared for several test cases in terms of sparsity preservation, accuracy, and convergence of the iterative method, GMRES. Direct methods with sparse techniques are very popular in the area of power system simulation. On the other hand, these types of methods are not suitable for large

Table 3. Iteration number in each NR step with the same preconditioner for every step. The test case is the Poland electricity system model with 2746 buses.

NR-iteration	ILU (0)		ILUT (0.001)		ILUT (1e-6)		2SP	
	Outer	Inner	Outer	Inner	Outer	Inner	Outer	Inner
1	29	2	7	5	1	3	2	2
2	39	10	7	6	1	4	1	9
3	43	5	7	1	1	4	1	9
4	37	4	7	7	1	4	1	9
5	-	-	-	-	1	4	-	-

problems due to fill-in and, more importantly, provide limited parallelism and memory requirements. Hence, in the area of power system simulation, iterative methods must be considered. To accelerate the convergence of the iterative methods one has to employ a preconditioner. The main idea of the proposed preconditioner is to reduce the number of eigenvalue clusters. To do this, integral form representations of the spectral projectors are used. The main computational cost of the proposed preconditioner is the numerical integration of the spectral projector via Gaussian quadrature. In order to benefit from the sparse structure of electrical power networks, the SPAI method can be employed to reduce the cost of the spectral projector computation. After obtaining the spectral projector, one needs to compute its left singular space to obtain the preconditioner. One can consider the issue as a sparse eigenvalue problem to reduce the computational cost. Moreover, it is always possible to employ partial eigendecomposition methods to obtain the left singular space of the spectral projector. These computations are performed only once and the same preconditioner can be used in subsequent NR iterations. In this regard, the proposed algorithm has an advantage over some well-known preconditioners, such as ILU. Computational tools and the structure of the suggested preconditioner are suitable for parallel processing. Therefore, our foresight about the parallel implementation of the suggested preconditioners will be effective and reliable. In future work, it is planned to improve the efficiency of the method with partial eigendecomposition methods and to implement it on a parallel platform to solve larger problems.

Nomenclature

A	Coefficient matrix
x	Solution
b	Right-hand-side vector
$\Lambda(A)$	Eigenvalue spectrum of matrix A
Sign (A)	Sign function of matrix A
I	Identity matrix
P (A)	Spectral projector of matrix A
Γ	Closed-curve containing the wanted eigenvalues in complex plane
a, b, c, d	Coordinate parameters for a rectangular Γ
K	Sparse approximate inverse of matrix A
ε	Sparsity parameter of SPAI algorithm
U	Left singular vectors of the projector P (A)
M	Preconditioner matrix
B_{ij}	Blocks of preconditioner matrix M
A_n	Coefficient matrix after applying preconditioner
b_n	Right-hand-side vector after applying preconditioner
x_n	Solution vector after applying preconditioner

References

[1] Kundur P. Power System Stability and Control. New York, NY, USA: McGraw-Hill: 1993.

[2] Dag H, Semlyen A. A new preconditioned conjugate gradient power flow. IEEE T Power Syst 2003: 18: 1248–1255.

[3] Boumediena L, Khiat M, Rahli M, Chaker A. Harmonic power flow in electric power systems using modified Newton–Raphson method. Int Rev Electr Eng 2009: 4: 410–416.

[4] Rashidinejad A, Habibi MR, Khorasani H, Rashidinejad M. A combinatorial approach for active and reactive power flow tracking. Int Rev Electr Eng 2010: 5: 2209–2216.

[5] Hojabri M, Hizam H, Mariun N, Abdullah SM, Pouresmaeil E. Available transfer capability for power system planning using Krylov-Algebraic method. Int Rev Electr Eng 2010: 5: 2251–2262.

[6] Golub G, Van Loan C. Matrix Computations. Baltimore, MD, USA: John Hopkins University Press, 1996.

[7] Freund R, Golub G, Nachtigal N. Iterative solution of linear systems. Act Num 1991; 1: 57–100.

[8] Dağ H, Alvarado FL. Computation-free preconditioners for the parallel solution of power system problems. IEEE T Power Syst 1997; 12: 585–591.

[9] Chen K. Matrix Preconditioning Techniques and Applications. Cambridge, UK: Cambridge University Press, 2005.

[10] Benzi M. Preconditioning techniques for large linear systems: a survey. J Comput Phys 2002: 182: 418–477.

[11] Dag H, Yetkin EF. A spectral divide and conquer method based preconditioner design for power flow analysis. In: IEEE 2010 International Conference on Power System Technology; 24–28 October 2010; Hangzhou, China. New York, NY, USA: IEEE. pp. 1–6.

[12] Dağ H, Yetkin EF, Manguoğlu M. A new preconditioner design based on spectral division for power flow analysis. Int Rev Electr Eng 2011; 6: 1339–1348.

[13] Hager WW. Applied Numerical Linear Algebra. Upper Saddle River, NJ, USA: Prentice Hall, 1988.

[14] Grote MJ, Huckle T. Parallel preconditioning with sparse approximate inverses. SIAM J Sci Comput 1997; 18: 838–853.

[15] Higham NJ. Functions of Matrices: Theory and Computation. Philadelphia, PA, USA: SIAM, 2008.

[16] Bai Z, Demmel J. Design of a Parallel Nonsymmetric Eigenroutine Toolbox. Part 1: Technical Report. Lexington, KY, USA: University of Kentucky, 1992.

[17] Kenney CS, Laub AJ. The matrix sign function. IEEE T Automat Contr 1995; 40: 1330–1348.

[18] Malyshev AN. Parallel algorithm for solving some spectral problems of linear algebra. Linear Algebra Appl 1993; 189: 489–520.

[19] Malyshev AN, Sadkane M. On parabolic and elliptic spectra dichotomy. SIAM J Matrix Anal A 1997; 18: 265–278.

[20] Kenney C, Laub AJ. Rational iteration methods for the matrix sign function. SIAM J Math Anal 1991; 21: 487–494.

[21] Kenney C, Laub AJ. On scaling Newton's method for polar decomposition and the matrix sign function. SIAM J Math Anal 1992; 13: 688–706.

[22] Koç ÇK, Bakkaloğlu B, Shieh LS. Computation of the matrix sign function using continued fraction expansion. IEEE T Automat Contr 1994; 39: 1644–1647.

[23] Barnard ST, Grote MJ. A block version of the SPAI preconditioner. In: Proceedings of the 9th SIAM Conference on Parallel Processing for Scientific Computing; 22–24 March 1999; San Antonio, TX, USA. Philadelphia, PA, USA: SIAM.

[24] Polizzi E. A density-matrix based algorithm for solving eigenvalue problems. Phys Rev B 2010; 79: 115–121.

[25] Wagner N. Spectral projectors and their applications in structural dynamics. Proc Appl Math Mech 2004; 4: 117–118.

[26] Benzi M. A comparative study of sparse approximate inverse preconditioners. Appl Numer Math 1999; 30: 305–340.

[27] Yetkin EF, Dağ H. Sparsity preserving computation for spectral projectors. AIP Conf Proc 2011; 1368: 267–271.

[28] Rommes J, Martins N. Exploiting structure in large-scale electrical circuits and power system problems. Linear Algebra Appl 2009; 401: 318–333.

[29] Saad Y. Iterative Methods for Linear Sparse Systems. 2nd ed. Philadelphia, PA, USA: SIAM, 2003.

[30] Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER's extensible optimal power flow architecture. In: IEEE 2009 Power and Energy Society General Meeting; 26–30 July 2009; Calgary, AB, Canada. New York, NY, USA: IEEE. pp. 1–7.

[31] Barrett R, Berry M, Chan TF, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, Van der Vorst H. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. 2nd ed. Philadelphia, PA, USA: SIAM, 1994.