Multiple-rank modification of symmetric eigenvalue problem

HyungSeon Oh*, Zhe Hu

Electrical Engineering, State University of New York at Buffalo, United States

Abstract

Rank-1 modifications applied \(k\)-times \((k > 1)\) often are performed to achieve a rank-\(k\) modification for enhancing computational efficiency. As the first step toward a rank- \(k\) modification, an algorithm to perform a rank-2 modification is proposed and tested. The computation cost of our proposed algorithm is in \(O(n^2)\) where \(n\) is the cardinality of the matrix of interest. We also propose a general rank-\(k\) update algorithm based on the Sturm Theorem, and compare our results to those of the direct eigenvalue decomposition and of a perturbation method.

© 2018 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

* Corresponding author.
E-mail address: hso1@buffalo.edu (H. Oh).
Method details

Optimal power flow plays a key role in the operation and planning studies for power systems. Due to its nonlinearity and non-convexity, a numerical solution is pursued using heuristic methods [1,2]. The most widely used method is a Lagrange relaxation, which relies on computationally expensive matrix factorizations. A matrix commonly involves a low-rank update; therefore, it would be a viable option to update the factors instead of the expensive re-factorization process to evaluate them. An LU modification method is first introduced, and many study results show its effectiveness [8,9]. While very relevant to power system studies because it can preserve the sparsity, the lack of numerically stability is an important issue. A stable update is achieved by updating Cholesky factorization [9–11]; however, its applicability is limited to positive semi-definite (PSD) matrices. The matrices associated with power systems are not, in general, PSD. Modifying eigenvalue/eigenvector pairs [12] would be a good candidate as they are numerically stable and can modify a large-scale matrix.

Consider a real symmetric matrix \( \mathbf{A} \in \mathbb{R}^{n \times n} \) with known eigenvalue decomposition \( \mathbf{A} = \mathbf{Q} \Lambda \mathbf{Q}^T \), to which a symmetric perturbation is added. \( \mathbf{Q} \) is the matrix comprised of eigenvectors and \( \Lambda \) is a diagonal eigenvalue matrix, i.e., \( \mathbf{Q} \) is the \( i \)th column vector \( q_i \) of \( \mathbf{Q} \), and \( \Lambda \) is the corresponding eigenvalue. If the perturbation is a rank-one matrix, \( \sigma \mathbf{v} \mathbf{v}^T \) where \( \sigma \) and \( \mathbf{v} \) are a scalar and a vector in \( \mathbb{R}^{n \times 1} \), the eigenvalue of new matrix \( \mathbf{A} + \sigma \mathbf{v} \mathbf{v}^T \) could be given by a “secular equation” [4]:

\[
f(\lambda) = 1 + \sigma \sum_{j=1}^{n} \frac{\zeta_j^2}{\xi_j - \lambda} = 0
\]

where \( \lambda_i \) is the \( i \)th eigenvalue of original matrix \( \mathbf{A} \), and \( \zeta_i \) is the \( i \)th element in vector \( \mathbf{z} = \mathbf{Q} \mathbf{v} \). If \( \zeta_i \) are all nonzero and \( \lambda_i \) are distinct, then this equation has \( n \) solutions. On interval \([\lambda_{n-1}, \lambda_{n+1}]\), function \( f \) is monotonic. These properties lead to several efficient and stable methods to solve the secular equation [4,5].

However, it is difficult to find the rank-1 modification matrix during the heuristic algorithm. In most situations, the perturbation is not a rank-1 matrix. A reduction of the rank in the perturbation matrix is possible by utilizing eigenvalue decomposition after selecting a subset of eigen-pairs. The disadvantages of this approach are: 1) computationally expensive eigenvalue decomposition, and 2) poor accuracy when a subset of eigenvalue pairs are included. It will be beneficial to modify high-rank eigenvalues directly. Eigenvalue decomposition is performed for a symmetric matrix, and the perturbation is made in preserving the symmetry. In an iterative method, binding constraint sets change, which involves an update such as \( e_i^T a + a^T e_i \) where \( a \) is a vector in \( \mathbb{R}^{n \times 1} \) and \( e_i \) is the \( i \)th column vector of the identity matrix in \( \mathbb{R}^{n \times n} \). Such a change is easily recognized without any further analysis to identify the perturbation vector as a rank-1 update process. This is the primary motivation for a rank-2 update.

In this paper, we propose a new, direct rank-\( k \) modification where \( \sqrt{n} \leq k \ll n \), i.e., \( k \) is small but not negligible such as \( k = 1 \) or 2. In Section Theory of eigenvalue updates, we present the theoretical background for the multiple-rank modification for rank-\( k \) updates. Section Numerical result lists the numerical results, and Section Conclusion outlines conclusions and future works.
Theory of eigenvalue updates

Secular equation of multiple-rank modification

The multiple-rank modification theory was carried out in [3]. Considering the eigenvalue decomposition of a real symmetric matrix $A + K K^T$, where $K \in \mathbb{R}^{n-k} \times \sqrt{n} \leq k \ll n$. The eigenvalue decomposition of matrix $A$ is known. It is pointed out that the solution of a multiple-rank modification problem is equivalent to roots of:

\[
f(\lambda) = \det \left[ I_k - K^T (\lambda I_n - A)^{-1} K \right] = \det \left[ I_k - U^T (\lambda I_n - A)^{-1} U \right]
\]

\[
= 1 + \sum_{r=1}^{k} (-1)^r \sum_{1 \leq k_1 \leq \ldots \leq k_r \leq k} \left[ \det \left( U_{j_1, j_2, \ldots, j_r} \right) \right] \frac{1}{\lambda - \lambda_j} \prod_{s \neq i} \left( \lambda_s - \lambda_i \right) \prod_{s \in \{j_1, \ldots, j_r\}} \frac{1}{\lambda_s - \lambda_i} \tag{1}
\]

where $U = Q^T K U_{j_1, j_2, \ldots, j_r} = (e_{i_1} \cdots e_{i_r})^T U (e_{j_1} \cdots e_{j_r})$, and $e_i$ is the $i$th column vector from an identity matrix [3]. Arbenz et al. point out that solving $f(\lambda) = 0$ is numerically difficult [3]. While calculating the value of determinants is expensive, rearranging it is possible in the following way:

\[
= \sum_{r=1}^{k} (-1)^r \sum_{1 \leq k_1 \leq \ldots \leq k_r \leq k} \left[ \det \left( U_{j_1, j_2, \ldots, j_r} \right) \right] \frac{1}{\lambda - \lambda_j} \prod_{s \neq i} \left( \lambda_s - \lambda_i \right) \prod_{s \in \{j_1, \ldots, j_r\}} \frac{1}{\lambda_s - \lambda_i} \tag{2}
\]

where $\alpha_i^{(0)} = \det \left[ \begin{array}{cc} 0 & e_i^T U \\ U^T e_i & I_r + U^T D_i U \end{array} \right]$ and $D_i = \text{diag} \left( \frac{1}{\lambda_{i-1} - \lambda_i}, \ldots, \frac{1}{\lambda_{n-1} - \lambda_i}, 0, \frac{1}{\lambda_{n+1} - \lambda_i}, \ldots, \frac{1}{\lambda_n - \lambda_i} \right)$.

Therefore, similar to a rank-1 modification, $f$ also can have a secular equation:

\[
f(\lambda) = 1 + \sum_{j=1}^{n} \alpha_j^{(0)} \frac{1}{\lambda_j - \lambda} \tag{3}
\]

This secular equation is called “multiple-rank secular equation”. The matrix determinant is $(k + 1) \times (k + 1)$ matrix. When $k \ll n$, formulating (3) will not increase the computation cost significantly; however, with increased $k$, the determinants calculation involves a heavy computation, making this approach inefficient.

Another advantage is that this method does not require an orthogonal perturbation matrix $V$. Therefore, an additional eigenvalue decomposition for the perturbation matrix is not necessary. However, solving multiple-rank secular equations is more computationally demanding than that of the rank-1 update process. In a rank-1 modification, $f$ is monotonically increasing on each subinterval $(\lambda_i, \lambda_{i+1})$. Therefore, only one root exists in the range of $[\lambda_i, \lambda_{i+1}]$. The multiple-rank secular function is
not monotonic on an interval of \((\lambda_i, \lambda_{i+1})\). There would be at least 0 and at most \(r\) solutions. Although multiple-rank modification problems have a similar secular equation to a rank-1, most of the methods used to solve rank-1 problems in [3] cannot be directly used for multiple-rank problems. To solve the multiple-rank secular equations, we formulate two sub-problems: the first sub-problem is intended to locate eigenvalues in checking how many roots are on each subinterval; the second finds the value for these roots on each subinterval. The details are listed in Section Numerical Result. We may not need to update all of the eigenvalues—if only the eigenvalues in some specific interval are of interest, then the first sub-problem is enough.

Location of eigenvalues

In the process for the rank-2 modification, eigenvalues are located according to the signs of the coefficients in a polynomial long division, \(\mu_i\). For the case of rank-\(k\), they become more complicated to locate. In general, the one-column-one-row update approximates the modification while preserving the sparsity and the symmetry [5]:

Step 1. \(A = A_0 + ab^T + ba^T = Q(\Lambda + xy^T + yx^T)Q^T\)

Step 2. \(\Lambda + xy^T + yx^T = \Lambda + (x\ y)\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}(x\ y)^T = \Lambda + (x\ y)S\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}S^T(x\ y)^T\)

where \(S\) is the Schur complement. Let \((u_1\ u_2) = (x\ y)S\), which yields \(\Lambda + xy^T + yx^T = \Lambda + u_1u_1^T - u_2u_2^T\). Therefore, if we can find the dominant column-row pair, it is acceptable to update eigenvalues using several rank-2 modifications.

Considering the secular equation of a rank-2 modification, \(A + \sigma_1v_1v_1^T + \sigma_2v_2v_2^T\) with \(\sigma_1 > 0\) and \(\sigma_2 > 0\). In this case, the values of all the eigenvalues increase, which is termed a double-right shift. \(R\) is divided into \(n + 1\) subintervals by original eigenvalues \(\{\lambda_1, \ldots, \lambda_n\}\). When the rank of the perturbation matrix is 2, there are 0, 1, or 2 roots within each subinterval. We define a location vector \(L \in Z^{(n+1) \times 1}\), where each element represents the number of roots within the corresponding subinterval. We can approximate locations of new eigenvalues using the Courant-Weyl principle [6]:

\[
\lambda_k \leq \lambda_{k}^{\text{max}}; \quad 1 \leq k \leq n \quad \text{and} \quad \lambda_k^{\text{max}} \leq \lambda_{k+2}; \quad 1 \leq k \leq n - 2
\]

Also, from the signs of \(\mu_i\), the parity of \(L\) could be determined.

If \(\mu_i \cdot \mu_{i+1} > 0\), there is one eigenvalue in \((\lambda_i, \lambda_{i+1})\); otherwise, there are either zero or two eigenvalues in this interval \((\lambda_i, \lambda_{i+1})\). These three conditions are already sufficient to locate new eigenvalues. Fig. 1 illustrates a pseudo-code of Algorithm 1a for the location vector.

Note that the condition of Algorithm 1a is \(\sigma_1 > 0\) and \(\sigma_2 > 0\). The condition can be generalized as either \(\sigma_1 < 0\) and \(\sigma_2 < 0\) (Algorithm 1b), or as \(\sigma_1 < 0\) and \(\sigma_2 > 0\) (Algorithm 1c). For the first case, instead of searching for a root from the first subinterval, we can start from the last subinterval. And for situation (c), the sign of \(\mu_1\) can determine the number of eigenvalues on the first interval. Fig. 2 shows the pseudo-code for these two conditions.

Algorithm 1a. Locating eigenvalues (double-right shift)

\[
M = 0; \quad L_1 = 0
\]

for \(i = 1: n\)

if \(\mu_i \cdot \mu_{i+1} > 0\)

\(L_{i+1} = 1; \quad M = M+1;\)

else if \(M+2 > i\)

\(L_{i+1} = 0;\)

else

\(L_{i+1} = 2; \quad M = M+2;\)

end

Fig. 1. Pseudo-code of formulating locating vector when the matrix is under double-right shift update.
Algorithm 1b. Locating eigenvalues (double-left shift)

\[ M = 0; \quad L_{n+1} = 0 \]

for \( i = 1:n \)
  if \( \mu_{i} \cdot \mu_{i+1} > 0 \)
    \( L_{i+1} = 1; \quad M = M + 1; \)
  else if \( M + 2 > n - i + 1 \)
    \( L_{i+1} = 0; \)
  else
    \( L_{i+1} = 2; \quad M = M + 2; \)
end

Algorithm 1c. Locating eigenvalues (one-right-one-left shift)

if \( \mu_{i} > 0 \)
  \( M = 0; \quad L_{1} = 0 \)
else
  \( M = 1; \quad L_{1} = 1 \)
end

for \( i = 1:n \)
  if \( \mu_{i} \cdot \mu_{i+1} > 0 \)
    \( L_{i+1} = 1; \quad M = M + 1; \)
  else if \( M + 2 > i \)
    \( L_{i+1} = 0; \)
  else
    \( L_{i+1} = 2; \quad M = M + 2; \)
end

Fig. 2. Pseudo-code of generating location vector when matrix is under double-left shift update and one-left-one-right shift update.

For a general rank-k modification, the sign of \( \mu_{i} \) is insufficient to calculate the location vector. We propose a modified Sturm Chain method as follows.

The original Sturm chain or Sturm sequence \([7]\) is a finite sequence of polynomials \( p_{0}(\lambda), p_{1}(\lambda), \ldots, p_{m}(\lambda) \) of a decreasing degree with the following properties:

- \( p_{0}(\lambda) \) is square free (no square factors, i.e., no repeated roots)
- \( p_{0}(\lambda) = 0 \), then \( \text{sign}(p'(\lambda)) = \text{sign}(p_{1}(\lambda)) \)
- If \( p_{i}(\lambda) = 0, 0 < i < m \), then \( \text{sign}(p_{i-1}(\lambda)) = -\text{sign}(p_{i+1}(\lambda)) \)
- \( p_{m}(\lambda) \) does not change its sign.

Then by observing the signs of \( p_{0}(a), p_{1}(a), \ldots, p_{m}(a) \), and \( p_{0}(b), p_{1}(b), \ldots, p_{m}(b) \), the number of roots of \( p_{0}(\lambda) \) within \((a, b)\) could be determined. Although the original Sturm Theorem is for a polynomial function, we found that it can also be implemented on a secular function. For our secular equation, we define that:

\[
\frac{f_{m}(\lambda)}{\pi_{m}(\lambda)} = \frac{p_{m}(\lambda)}{\pi_{m}(\lambda)} = c_{m} \sum_{j=1}^{n} \frac{\alpha_{j}^{(m)}}{\lambda - \lambda_{j}} \quad \text{where} \quad \pi_{m}(\lambda) = \prod_{j=1}^{n} (\lambda - \lambda_{j})
\]

(4)

The \( c_{m} \) is the quotient and \( \alpha_{j}^{(m)} \) is the scalar coefficients of the remainder associated with \((\lambda_{j} - \lambda)\), which can be evaluated from \( p_{m}(\lambda) \) and \( \pi_{m}(\lambda) \). Here we describe how we compute \( c_{m} \) and \( \alpha_{j}^{(m)} \) directly without evaluating \( p_{m}(\lambda) \) and \( \pi_{m}(\lambda) \). To expand the Sturm sequence of a non-polynomial secular equation, \( f_{0}(\lambda) \), we multiply the product of all \( n \) denominators in the summation in (4). Then we expand the \( m^{\text{th}} \) Sturm sequence and divide a proper product to make a similar form as in (4). Note that no eigenvalues are repeated. For \( f_{1}(\lambda) \), to satisfy the second condition, we need to take the derivative of
\[ f_0(\lambda). \] Direct derivation of \( f_0(\lambda) \) will lead to second-order terms. Instead, take the derivative of the following function:

\[
p_0(\lambda) = \pi_0(\lambda) \left( 1 - \sum_{j=1}^{n} \frac{\alpha_j^{(0)}}{\lambda - \lambda_j} \right)
\]

Then we have:

\[
p_1(\lambda) = \frac{dp_0(\lambda)}{d\lambda} = \frac{df_0(\lambda)}{d\lambda} \pi_0(\lambda) + f_0(\lambda) \left( \sum_{j=1}^{n} \frac{1}{\lambda - \lambda_j} \right) \pi_0(\lambda)
\]

\[
= \left\{ \sum_{j=1}^{n} \frac{\alpha_j^{(0)}}{\lambda - \lambda_j} \lambda_j + \sum_{j=1}^{n} \frac{1}{\lambda - \lambda_j} \right\} \pi_0(\lambda)
\]

\[
= \left\{ \sum_{j=1}^{n} \frac{\alpha_j^{(0)}}{\lambda - \lambda_j} \lambda_j + \sum_{j=1}^{n} \frac{1}{\lambda - \lambda_j} \right\} \pi_0(\lambda)
\]

\[
= \left[ c_1 - \sum_{j=1}^{n} \frac{\alpha_j^{(1)}}{\lambda - \lambda_j^2} \right] \pi_1(\lambda) \text{where} \mu_j^{(1)} = 1 - \sum_{i=1}^{n} \frac{\alpha_i^{(0)} + \alpha_j^{(0)}}{\lambda_j - \lambda_i} \quad \text{and} \quad \alpha_j^{(1)} = (\lambda_n - \lambda_j) \mu_j^{(1)}
\]

Here \( \alpha_j^{(1)} \) is not related to \( \lambda \). The Sturm chain theorem finds the number of solutions inside a given range to satisfy \( p_0(\lambda) = 0 \). However, the direct application of the theorem to \( p_0(\lambda) \) is not practical because the computation of the coefficients is numerically unstable. Inspired by the fact that only the change in the signs of \( p_i(\lambda) \) is important, which is the product between \( \pi_i(\lambda) \) and \( f_i(\lambda) \), we propose a modified Sturm series to locate the updated eigenvalues. For this purpose, \( p_2(\lambda) \ldots p_n(\lambda) \) are generated by long division used for polynomial functions that can be also expanded to a secular function.

### Secular Long Division

For secular functions with same eigenvalues \( d \),

\[
p_n(\lambda) = c_m \sum_{j=1}^{n-1} \frac{\alpha_j^{[m-1]}}{\lambda - d_j} \prod_{k=1}^{j} (x - d_k) \quad \text{and} \quad p_{m-1}(\lambda) = c_{m-1} \sum_{j=1}^{n-1} \frac{\alpha_j^{[m-1]}}{\lambda - d_j} \prod_{k=1}^{j} (x - d_k)
\]

\[
p_m(\lambda) = \sum_{j=1}^{m-2} \frac{\alpha_j^{[m-2]}}{\lambda - d_j} \prod_{j=1}^{m-2} (x - d_k) \quad \text{there exists secular long division}
\]

\[
f_m(\lambda) \pi_m(\lambda) - f_{m-1}(\lambda) A_m(\lambda - B_m) \pi_{m-1}(\lambda) + f_m(\lambda) \pi_m(\lambda) = 0
\]

\[
A_m = \frac{c_{m+2}}{c_{m-1}} \quad \text{and} \quad B_m = -d_{m+2} - \sum_{j=1}^{m-2} \frac{\alpha_j^{[m-2]}}{c_{m-2}} - \sum_{j=1}^{m-1} \frac{\alpha_j^{[m-1]}}{c_{m-1}} \text{(See Appendix)}
\]

In the Sturm chain theorem, the number of changes in the sign of the polynomial functions \( p_0(\lambda) \ldots p_n(\lambda) \) equals the number of solutions inside a region. In \([-\infty, \infty] \), the signs of \( f_0(\lambda) \ldots f_n(\lambda) \) are same as those of \( c_0 \ldots c_n \), the signs of \( \pi_0(\infty) \ldots \pi_n(\infty) \) are all positive, and those of \( \pi_0(-\infty) \ldots \pi_n(-\infty) \) are all negative. As a result, the solutions that satisfy \( p_0(\lambda) = 0 \) are the number of positives in \( c_0 \ldots c_n \) where \( c_0 = 1 \). In the computation process of \( c_0 \ldots c_n \), the floating number can result in a change in the sign of \( c_0 \ldots c_n \). If one terminates the Secular long division when it hits the non-positive \( c \) for the first time, the partial Sturm series yields the lower bound of the solutions between \([-\infty, \infty] \). Furthermore, one can break down the regions where at least one solution exists, yielding a solution as in Section Algorithm to find solutions for the secular equation. Suppose \( N_0^n + 1 \) \( c \) is positive at the first Sturm series, and accordingly \( N_0^n \) solutions \( \xi \)s are identified. Then a new
polynomial function $p_0^1(\lambda)$ is defined as follows:

$$p_0(\lambda) = \left[ 1 - \sum_{j=1}^{n} \frac{a_j^{(0)}}{\lambda - \lambda_j} \right] \prod_{j=1}^{n} (\lambda - \lambda_j) = p_0^1(\lambda) \prod_{j=1}^{N_0} (\lambda - \xi_j) \rightarrow p_0^1(\lambda)$$

$$= \left[ 1 - \sum_{j=1}^{n} \frac{a_j^{(0)}}{\lambda - \lambda_j} \right] \frac{\prod_{j=1}^{n} (\lambda - \lambda_j)}{\prod_{s=1}^{N_0} (\lambda - \xi_s)}$$

By multiplying the first term after the square parenthesis, (7) becomes:

$$p_0^1(\lambda) = \left[ 1 - \sum_{j=1}^{n} \frac{a_j^{(0)}}{\lambda - \lambda_j} \right] \frac{\prod_{j=1}^{n} (\lambda - \lambda_j)}{\prod_{s=1}^{N_0} (\lambda - \xi_s)}$$

$$= \left[ 1 - \sum_{j=1}^{n} \left( \frac{\lambda_j - \lambda_1}{\lambda_j - \xi_1} \right) \frac{\prod_{j=1}^{n} (\lambda - \lambda_j)}{\prod_{s=1}^{N_0} (\lambda - \xi_s)} \right] \prod_{j=1}^{n} (\lambda - \lambda_j)$$

Since $\xi_1$ is a solution to $p_0(\lambda) = 0$ (i.e., $p_0(\xi_1) = 0$), then $\pi_0(\xi_1) \neq 0$, and $1 - \sum_{j=1}^{n} a_j^{(0)}$ vanishes. After multiplying all $N_0$ terms, $p_0^1(\lambda)$ becomes:

$$p_0^1(\lambda) = \left[ 1 - \sum_{j=1}^{n} \frac{\beta_j^{(0)}}{\lambda - \lambda_j} \right] \prod_{j=N_0+1}^{n} (\lambda - \lambda_j)$$

where $\beta_j^{(0)} = \left[ \prod_{p=1}^{N_0} \frac{d_j - d_p}{d_j - \xi_p} \right] a_j^{(0)}$

The formula for $p_0^1(\lambda)$ resembles that for $p_0(\lambda)$, which allows the Sturm series to expand with fewer terms by $N_0$. Since the first $c$ in $p_0^1(\lambda)$ is positive, it is guaranteed to have at least one solution using the new Sturm series. This process will continue until all $n$ solutions are identified.

**Algorithm to find solutions for the secular equation**

Different from the rank-1 modification, the secular equation of rank-2 is not monotonically increasing at each interval. Therefore, most of the algorithms for a rank-1 update may not converge. The Divide-and-Conquer (D&C) method efficiently addresses this problem [13]. As long as the location vector is formulated, the D&C is fast and parallelizable. The strategy to find the roots to the secular equation is listed in the D&C zero finder. Fig. 3 is an example of a rank-2 secular equation. In the problem, the signs of the coefficients are mixed, i.e. positive and negative. This is usually observed when both updating and downdating are performed in a rank-2 update. We would like to find where the roots exist to apply for the D&C zero finder. Algorithm 1c provides an efficient way to locate the root-termed location vector. Since it is a rank-2 update between two eigenvalues, 0, 1, or 2 roots can exist. Algorithm 1c utilizes the fact that, as shown in Fig. 3, $(-\infty, 0), (0, 1), (1, 1.5), (1.5, 3), (3, 4)$, and $(4, \infty)$, at most two roots exist.

With a location vector, the D&C method can be utilized.
Algorithm to update eigenvectors

Once an eigenvalue is computed, the corresponding eigenvector \( v \) can be updated using the relationship between an eigenvalue and an eigenvector, i.e., \( (\Lambda + LL^T)u = \lambda u \) where \( L = Q^T K \) and \( u = Q^T v \). Solving the relationship for \( u \) yields \( u = (\Lambda - \lambda I)^{-1}L\alpha \) where \( \alpha = -L^Tu, \alpha \in \mathbb{R}^k \), which implies the eigenvector is in the column space spanned by the rank-\( k \) update \( K \). \( \alpha \) is in the simple-article space of \( \{I + L^T(\Lambda - \lambda I)^{-1}L\}_{k \times k} \), and let \( \alpha_0 \) denote the simple-article space vector. The row-rank update \( k \) where \( \sqrt{n} \leq k \ll n \) makes finding the simple-article space computationally inexpensive. Since the eigenvector is a unit vector, \( v \) becomes:

\[
v = \frac{Q(\Lambda - \lambda I)^{-1}Q^T K\alpha_0}{\| Q(\Lambda - \lambda I)^{-1}Q^T K\alpha_0 \|_2}
\]  

(10)

Numerical result

In the D&C method, the tolerance of the error in a root is set to \( 10^{-9} \) so that the precision of the solution is within a numerical error range. Fig. 4 illustrates the comparison of the computation time of a rank-1 update, a rank-2 update, and the original eigenvalue decomposition. For the numerical computation to perform direct eigenvalue decomposition, we used the MATLAB eig function [14]. The testing results shows the order \( p \) of the computation are in \( O(n^p) \):

- Two times rank-1 update: \( p = 1.81 \)
- Rank-2 update: \( p = 1.45 \)
Fig. 4. Computation time of two times rank-1 update, rank-2 update, and original eigenvalue decomposition.

Fig. 5. Comparison of the computation time for Rank-3 update of the proposed method and the original MATLAB eig function.
Direct eigenvalue decomposition with MATLAB: \( p = 2.88 \)

The computation cost of a rank-2 modification is significantly better than that of a rank-1 modification. Parallel computation will further enhance the computational efficiency.

In Fig. 5, the red line is the computation time for a randomly generated rank-3 update, and the blue line is that of original eigenvalue decomposition. Updating the eigenvalues by the secular Sturm Chain method is in \( O(n^{1.95}) \), which is more expensive than a rank-2 update; however, it is still much more efficient than the original eigenvalue decomposition.

When the perturbed matrix \( K \) is small, i.e., \( \text{norm}(K) << 1 \), the eigenvalue and eigenvector pairs can be estimated from:

\[
\lambda_j = \lambda_j^0 + \left( K^T v_j^0 \right)^T \left( K^T v_j^0 \right) \quad \text{and} \quad v_j = v_j^0 + \sum_{i=1, i \neq j}^{n} \frac{\left( K^T v_i^0 \right)^T \left( K^T v_i^0 \right)}{\lambda_i^0 - \lambda_j^0} v_j^0
\]

(11)

However, as the norm increases, the perturbation method starts to fail as its assumption to ignore higher order terms is invalid. Fig. 7 shows the comparison where the norm(\( K \)) is 0.3 and \( n = 100 \). While the performance of the proposed method does not change with the norm of \( K \), the eigenvalues, the
orthogonality, and the product between eigenvalue and eigenvector pairs all deviate from the desired quantities.

Fig. 6 illustrates the comparisons in the results among direct eigenvalue decomposition, the proposed method, and the perturbation method when the norm($K$) is 0.01 and $n = 100$. When the matrix $K$ is very small, the perturbation method [15] provides a reasonably close estimate of the eigenvalue and eigenvector pair. Fig. 7 shows that the error of the perturbation method [15] can be large in estimating the eigenvalue and the eigenvector when the norm($K$) is not insignificant. The proposed method always provides a precise estimation for both eigen-pairs. While the proposed method is efficient when the rank of the matrix $K$ is small (approximately 10% of $n$) but greater than $\sqrt{n}$, the computation time increases as the rank of $K$ increases. The proposed method is more efficient than any other to compute a subset of eigenvectors when $\sqrt{n} \leq k < n$. For example, a simple-article space is useful in many engineering fields, and the identification of an entire eigenvector space may not be necessary. In such a circumstance, the eigenvalues are all zeros and the computation time to find eigenvectors is still much smaller than that of direct eigenvalue decomposition (compare the dotted blue line and the black solid line in (D) from Figs. 6 and 7). For the consideration of computation time, the computation of the Sturm series increases in $\mathcal{O}(n^2)$ because the major computation task of the proposed method is to compute the eigenvalue of $I_k - U^T (\lambda I_n - A)^{-1} U$. In comparison, the direct eigenvalue decomposition of the matrix takes in $\mathcal{O}(nk^2)$. For a given value of $k$, the computation time
grows in quadratic (See Fig. 8 for the results where $k = 40–60$ for various values of $n$). The values for $p$ in $\mathcal{O}(n^p)$ decrease with increasing $k$ (2.45, 2.19, and 2.16 for $k = 40, 50, \text{and} 60$, respectively).

### Conclusion

In this paper, our new algorithm for a rank-$k$ modification of eigenvalue decomposition is presented. Computation performance is significantly improved in comparison to rank-1 modification methods. We propose a method relying on the location vector, which is tested on several systems and results show our proposed method is efficient. When there is a topology change in a network, market power is greatly affected and is reflected in the symmetric dispatch sensitivity to price. We plan to apply this algorithm to update market power computation associated with topological control [16]. Because a topological change affects many but not all locations, the condition ($\sqrt{n} \leq k \ll n$) is satisfied where $k$ and $n$ are the numbers of the affected substations and of all the substations, respectively.

### Acknowledgment

This work is supported by the U.S. Depart of Energy through the CERTS (Consortium for Electric Reliability Technology Solutions) program, PO# 7069673.
Appendix A

Suppose there exists a secular long division series as follows:

\[ p_{m-2}(x) - A_m(x - B_m)p_{m-1}(x) + p_m(x) = 0 \]

where

\[
\begin{aligned}
    p_m(x) &= \left( c_m - \sum_{j=1}^{n-m} \alpha_j^{[m]} \right) \prod_{k=1}^{n-m} (x - d_k) \\
    p_{m-1}(x) &= \left( c_{m-1} - \sum_{j=1}^{n-m+1} \alpha_j^{[m-1]} \right) \prod_{k=1}^{n-m+1} (x - d_k) \\
    p_{m-2}(x) &= \left( c_{m-2} - \sum_{j=1}^{n-m+2} \alpha_j^{[m-2]} \right) \prod_{k=1}^{n-m+2} (x - d_k)
\end{aligned}
\]  

(A1)

By dividing (A1) by \( c_{m-2} \prod_{k=1}^{n-m+2} (x - d_k) \), one finds

\[
1 - \sum_{j=1}^{n-m+2} \beta_j^{[m-2]} \frac{1}{x - d_j} + h_m \left(1 - \sum_{j=1}^{n-m+1} \beta_j^{[m-1]} \frac{1}{x - d_j} + \frac{B_m - d_{n-m+2}}{x - d_{n-m+2}} - \frac{B_m - d_{n-m+2}}{x - d_{n-m+2}} \sum_{j=1}^{n-m+1} \frac{1}{x - d_j} \right) h_m = 0
\]  

(A2)

where \( g_m = \frac{c_m}{c_{m-2}}, h_m = \frac{c_{m-1}}{c_{m-2}} A_m, \beta_j^{[m-2]} = \frac{\alpha_j^{[m-2]}}{c_{m-2}}, \beta_j^{[m-1]} = \frac{\alpha_j^{[m-1]}}{c_{m-1}}, \) and \( \gamma_j^{[m]} = \frac{\alpha_j^{[m]}}{c_{m-2}}. \)

Setting the constant term zero leads to \( h_m = 1 \), i.e., \( A_m = \frac{c_{m-1}}{c_{m-2}} \). Rearranging (A2) yields,

\[
- \sum_{j=1}^{n-m+2} \beta_j^{[m-2]} \frac{1}{x - d_j} + \sum_{j=1}^{n-m+1} \beta_j^{[m-1]} \frac{1}{x - d_j} + \frac{B_m - d_{n-m+2}}{x - d_{n-m+2}} - \frac{B_m - d_{n-m+2}}{x - d_{n-m+2}} \sum_{j=1}^{n-m+1} \frac{1}{x - d_j} = 0
\]  

(A3)

The product of fractions can be expanded as follows:

\[
\begin{aligned}
\prod_{j=1}^{n-m+2} \frac{1}{x - d_j} &= \frac{1}{d_{n-m+2}} - \frac{d_{n-m+2}}{d_{n-m+2}} x - \frac{1}{x - d_{n-m+2}} \\
\prod_{j=1}^{n-m+1} \frac{1}{x - d_j} &= \frac{1}{d_{n-m+1}} - \frac{d_{n-m+1}}{d_{n-m+1}} x - \frac{1}{x - d_{n-m+1}}
\end{aligned}
\]  

(A4)

From (A3) and (A4), one finds:
\[
\sum_{j=1}^{n-m} \left[ \beta_j^{(m-1)} - \beta_j^{(m-2)} \right] \frac{1}{x - d_j} + \left[ B_m - d_{n-m+2} - \beta_j^{(m-2)} \right] \frac{1}{x - d_{n-m+2}} + \left[ \gamma_j^{(m)} \right] \frac{1}{x - d_{n-m+1}} + \left[ \beta_j^{(m-1)} - \beta_j^{(m-2)} \right] \frac{1}{x - d_{n-m+1}} + \gamma_j^{(m)} \frac{1}{x - d_j} 
\]

\[
+ \left[ B_m - d_{n-m+2} - \beta_j^{(m-2)} \right] \frac{1}{x - d_{n-m+1}} + g_m 
\]

\[
+ (B_m - d_{n-m+2}) \frac{1}{x - d_{n-m+1}} \sum_{i=1}^{n-m-1} \frac{1}{d_{n-m+2} - d_j} + \sum_{i=1}^{n-m} \frac{1}{d_{n-m+2} - d_j} 
\]

\[
\times \frac{1}{d_{n-m+2} - d_{n-m+1}} \frac{1}{x - d_{n-m+1}} = 0 \tag{A5}
\]

Setting all the constants of \( x - d_j \) in (A5) zero yields:

\[
B_m = d_{n-m+2} + \sum_{j=1}^{n-m+2} \beta_j^{(m-2)} - \sum_{j=1}^{n-m+1} \beta_j^{(m-1)} 
\]

\[
\gamma_j^{(m)} = \left( d_{n-m+1} - d_j \right) \left[ -(d_{n-m+2} - d_j) \beta_j^{(m-2)} + \beta_j^{(m-1)} \left( B_m - d_j \right) \right] \tag{A6}
\]

\[
g_m = B_m \sum_{j=1}^{n-m+1} \beta_j^{(m-1)} - \sum_{j=1}^{n-m+2} \beta_j^{(m-2)} \left( d_{n-m+2} - d_j \right) - \sum_{j=1}^{n-m+1} \beta_j^{(m-1)} \left( d_{n-m+2} - d_j \right) 
\]

With (A6), it is possible to formulate:

\[
f_m(x) = \frac{p_m(x)}{\prod_{k=1}^{n-m} (x - d_k)} = c_m \sum_{j=1}^{n-m} \frac{\alpha_j^{(m)}}{x - d_j} \tag{A7}
\]

where \( c_m = c_m - 2g_m \) and \( \alpha_j^{(m)} = c_m - 2\gamma_j^{(m)} \).

References

[1] W. Ongsakul, P. Bhasaputra, Optimal power flow with FACTS devices by hybrid TS/SA approach, Int. J. Electr. Power Energy Syst. 24 (10) (2002) 851–857.

[2] R.D. Zimmerman, C.E. Murillo-Sánchez, R.J. Thomas, MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education, IEEE Trans. Power Syst. 26 (1) (2011) 12–19.

[3] P. Arbenz, W. Gander, G.H. Golub, Restricted rank modification of the symmetric eigenvalue problem: theoretical considerations, Linear Algebra Appl. 104 (1988) 75–95.

[4] J.R. Bunch, C.P. Nielsen, D.C. Sorensen, Rank-one modification of the symmetric eigenproblem, Numerische Mathematik 31 (1) (1978) 31–48.

[5] A.A. Melman, Numerical comparison of methods for solving secular equations, J. Comput. Appl. Math. 86 (1) (1997) 237–249.

[6] H.F. Weinberger, Vibrational Methods for Eigenvalue Approximation, Society for Industrial and Applied Mathematics, Philadelphia, 1974.

[7] D.G. Hook, P.R. McAree, Using Sturm sequences to bracket real roots of polynomial equations, Graphics Gems, Academic Press Professional Inc., 1990, pp. 416–422.

[8] L.M. Suhl, U.H. Suhl, A fast LU update for linear programming, Ann. Oper. Res. 43 (1) (1993) 33–47.

[9] P.E. Gill, G.H. Golub, W. Murray, et al., Methods for modifying matrix factorizations, Math. Comput. 28 (126) (1974) 505–535.

[10] R.B. Schnabel, E. Eskow, A new modified Cholesky factorization, SIAM J. Sci. Stat. Comput. 11 (6) (1990) 1136–1158.

[11] M. Seeger, Low Rank Updates for the Cholesky Decomposition, University of California at Berkeley, Tech. Rep., 2007.

[12] G.H. Golub, Some modified matrix eigenvalue problems, Siam Rev. 15 (2) (1973) 318–334.

[13] S. Skiena, Matrix multiplication, The Algorithm Design Manual, Springer-Verlag, Berlin, New York, 1998.
[14] Mathworks, Eigenvalues and eigenvectors. (2017) [Online] Available at https://www.mathworks.com/help/matlab/ref/eig.html.

[15] A. Hegde, J.C. Principe, D. Erdogmus, et al., Perturbation-based eigenvector updates for on-line principal components analysis and canonical correlation analysis, J. VLSI Process. Syst. Signal Image Video Technol. 45 (1–2) (2006) 85–95.

[16] G. Poyrazoglu, H. Oh, Optimal topology control with physical power flow constraints and N-1 contingency criterion, IEEE Trans. Power Syst. 30 (6) (2015) 3063–3071.