International Journal of Applied Mathematics

Volume 37 No. 5 2024, 511–521
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v37i5.3

COMPUTATIONAL METHOD FOR FIRST THREE DOMINANT EIGENMODES OF SYMMETRIC MATRICES

Pravin Singh [1] , Virath Singh [2,§] , Shivani Singh [3]

[1] University of KwaZulu-Natal
Private Bag X54001
Durban - 4001, SOUTH AFRICA
e-mail: singhprook@gmail.com

[2,§] University of KwaZulu-Natal
Private Bag X54001
Durban - 4001, SOUTH AFRICA
e-mail: singhv@ukzn.ac.za

[3] University of South Africa
Department of Decision Sciences
P.O. Box 392, Pretoria - 0003, SOUTH AFRICA

Abstract

In this paper we advocate a new method to compute the first three dominant eigenmodes of real symmetric matrices. Our method avoids deflation and can even compute the second and third mode, bypassing the need to compute the first mode.

Math. Subject Classification: 15A18, 15A45, 65F35

Key Words and Phrases: eigenvalues; symmetric matrix; computation

Received: February 1, 2024

1. **Introduction**

Eigenvectors and eigenvalues inherently feature in almost all spheres of the sciences. In mathematics there are many consequences of the fact that the eigenvectors form a canonical basis for the representation of a linear operator. The ratio of the dominant to least dominant eigenvalues of a symmetric linear system, determines the conditioning of that system [9]. The diagonalizability of symmetric matrices provides a means of solving a coupled system of linear differential equations. The nature of the eigenvalues of the Jacobian matrix determines the stability of the latter system. As there are numerous applications of eigenvalue theory, we shall focus on some recent ones only. Eigenmodes are used in data representation, operations on data as well as in the training of machine learning models [12]. They are indispensable in the development of algorithms for robotics [16], facial recognition [17] and language processing [18]. For example in principal component analysis, the dominant mode of the covariance matrix of a data set, is used to reduce the dimension of the problem [13]. Image compression using only few dominant eigenmodes retains most of the information pertaining to the image [14]. Google serves up web pages according to the PageRank score, which depends on the principal eigenvector of the Google matrix [2]. Recently the eigenvalues of a matrix was used to analyse peak viral load of the SARS-Cov2 virus [1]. The spread $\mathrm{sp}(\mathbf{A})$ of the adjacency matrix is useful in graph theory and its applications [3]. When most of the eigenvalues of the cryptocurrency correlation matrix lie outside the interval $(\lambda_{min}, \lambda_{max})$, where λ_{min}, λ_{max} represent the minimum and maximum eigenvalues of a random symmetric matrix of the same order, then there is good correlation between the respective cryptocurrencies. In quantum mechanics, the eigenvalues of the Fock matrix represent molecular orbital energies [15]. Iterative methods like the QR algorithm [19] is efficient and stable to compute all the eigenvalues. For sparse Hermitian matrices the Lanczos algorithm is preferred [8]. For large matrices, when few of the dominant modes are required, it is not feasible to use these methods. Bounding the spectrum $\sigma(\mathbf{A})$ of symmetric matrices is important in first attempting to locate the eigenvalues. These have been discussed extensively, using both the trace $\mathrm{tr}(\mathbf{A})$ and projections [4, 5, 6, 7]. We shall show in this article how it is possible to determine the first three dominant modes of a real symmetric matrix, without using deflation [10]. We use Julia Version 1.10.0 with nlsove for our computations.

2. **Theory**

Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ have $N \leq n$ distinct eigenvalues $\lambda_i, i = 1, 2, \ldots N$, arranged such that

$$|\lambda_1| > |\lambda_2| > |\lambda_3| > \cdots |\lambda_{N-1}| > |\lambda_N|. \tag{1}$$

We shall denote the algebraic multiplicity of λ_i by m_i. Since $\mathbb{R}^n = \oplus_{i=1}^N N(\mathbf{A}-\lambda_i\mathbf{I})$, where $N(\mathbf{A}-\lambda_i\mathbf{I})$ denotes the nullspace of $\mathbf{A}-\lambda_i\mathbf{I}$, it follows that arbitrary $\mathbf{x}_0 \in \mathbb{R}^n$, $\|\mathbf{x}_0\| = 1$, may be expressed uniquely by

$$\mathbf{x}_0 = \sum_{i=1}^{3} c_i\mathbf{u}_i + \mathbf{e}, \tag{2}$$

where $\{\mathbf{u}_i\}_{i=1}^N$, $\|\mathbf{u}_i\| = 1$ denotes a set of normalized mutually orthogonal eigenvectors corresponding to $\{\lambda_i\}_{i=1}^N$ and

$$\mathbf{e} = \sum_{i=4}^{N} c_i\mathbf{u}_i. \tag{3}$$

Then

$$\mathbf{A}^k\mathbf{x}_0 = \sum_{i=1}^{3} c_i\lambda_i^k\mathbf{u}_i + \mathbf{A}^k\mathbf{e} \tag{4}$$

$$\langle\mathbf{A}^k\mathbf{x}_0, \mathbf{x}_0\rangle = \sum_{i=1}^{3} c_i^2\lambda_i^k\mathbf{u}_i + \langle\mathbf{A}^k\mathbf{e}, \mathbf{e}\rangle. \tag{5}$$

Let $\alpha_k = \langle\mathbf{A}^k\mathbf{x}_0, \mathbf{x}_0\rangle$,$d_i = c_i^2$ and $\mathbf{e}_k = \langle\mathbf{A}^k\mathbf{e}, \mathbf{e}\rangle$, then we may write

$$\alpha_k = \sum_{i=1}^{3} d_i\lambda_i^k + e_k, \tag{6}$$

$$\alpha_{k+1} = \sum_{i=1}^{3} d_i\lambda_i^{k+1} + e_{k+1}. \tag{7}$$

$$\tag{8}$$

Let $S = \{1, 2, 3\}$ and choose $p \in S$. Multiply (6) by λ_p and subtract from (7) to obtain

$$\alpha_{k+1} - \lambda_p\alpha_k = \sum_{\substack{i=1\\ i\neq p}}^{3} d_i(\lambda_i - \lambda_p)\lambda_i^k + e_{k+1} - \lambda_p e_k. \tag{9}$$

Replacing k by $k+1$ in (9) results in

$$\alpha_{k+2} - \lambda_p\alpha_{k+1} = \sum_{\substack{i=1\\ i\neq p}}^{3} d_i(\lambda_i - \lambda_p)\lambda_i^{k+1} + e_{k+2} - \lambda_p e_{k+1}. \tag{10}$$

Choose $q \in S$, $q \neq p$, multiply (9) by λ_q and subtract from (10) to obtain

$$\alpha_{k+2} - (\lambda_p + \lambda_q)\alpha_{k+1} + \lambda_p\lambda_q\alpha_k \tag{11}$$

$$= d_r(\lambda_r - \lambda_p)(\lambda_r - \lambda_q)\lambda_r^k + e_{k+2} - (\lambda_p + \lambda_q)e_{k+1} + \lambda_p\lambda_q e_k, \tag{12}$$

where $r \in S$, $r \neq p, q$. Thus (12) simplifies to

$$\alpha_{k+2} - (\lambda_p + \lambda_q)(\alpha_{k+1} - e_{k+1}) + \lambda_p\lambda_q(\alpha_k - e_k) \\ = d_r(\lambda_r - \lambda_p)(\lambda_r - \lambda_q)\lambda_r^k. \tag{13}$$

Replacing k by $k+1$ in (13), we have

$$\alpha_{k+3} - (\lambda_p + \lambda_q)(\alpha_{k+2} - e_{k+2}) + \lambda_p\lambda_q(\alpha_{k+1} - e_{k+1}) \\ = d_r(\lambda_r - \lambda_p)(\lambda_r - \lambda_q)\lambda_r^{k+1}. \tag{14}$$

Taking the ratio of (14) and (13) yields

$$\lambda_r = \frac{\alpha_{k+3} - (\lambda_p + \lambda_q)(\alpha_{k+2} - e_{k+2}) + \lambda_p\lambda_q(\alpha_{k+1} - e_{k+1})}{\alpha_{k+2} - (\lambda_p + \lambda_q)(\alpha_{k+1} - e_{k+1}) + \lambda_p\lambda_q(\alpha_k - e_k)}. \tag{15}$$

Now $e_k = \langle \mathbf{A}^k\mathbf{e}, \mathbf{e}\rangle = \mathcal{O}(\lambda_4^k)$, and $\alpha_k = \mathcal{O}(\lambda_1^k)$, thus $\frac{|e_k|}{|\alpha_k|} = \mathcal{O}(|\frac{\lambda_4}{\lambda_1}|^k) << 1$ for k large, thus we shall ignore all $e_{k+i}, i = 0, 1, 2, 3$ terms in (15). Also replacing k by $k+1$ in the resulting equation yields

$$\lambda_r = \frac{\alpha_{k+3} - (\lambda_p + \lambda_q)\alpha_{k+2} + \lambda_p\lambda_q\alpha_{k+1}}{\alpha_{k+2} - (\lambda_p + \lambda_q)\alpha_{k+1} + \lambda_p\lambda_q\alpha_k} \tag{16}$$

$$= \frac{\alpha_{k+4} - (\lambda_p + \lambda_q)\alpha_{k+3} + \lambda_p\lambda_q\alpha_{k+2}}{\alpha_{k+3} - (\lambda_p + \lambda_q)\alpha_{k+2} + \lambda_p\lambda_q\alpha_{k+1}}. \tag{17}$$

Equation (17) is written as

$$[\alpha_{k+3} - (\lambda_p + \lambda_q)\alpha_{k+2} + \lambda_p\lambda_q\alpha_{k+1}]^2 \\ = [\alpha_{k+2} - (\lambda_p + \lambda_q)\alpha_{k+1} + \lambda_p\lambda_q\alpha_k] \\ \times [\alpha_{k+4} - (\lambda_p + \lambda_q)\alpha_{k+3} + \lambda_p\lambda_q\alpha_{k+2}]. \tag{18}$$

Equation (18) may be simplified to yield

$$a_k^{(0)} + a_k^{(1)}(\lambda_p + \lambda_q)^2 + a_k^{(2)}\lambda_p^2\lambda_q^2 + a_k^{(3)}(\lambda_p + \lambda_q) \\ + a_k^{(4)}\lambda_p\lambda_q + a_k^{(5)}\lambda_p\lambda_q(\lambda_p + \lambda_q) = 0, \tag{19}$$

where

$$a_k^{(0)} = \alpha_{k+3}^2 - \alpha_{k+2}\alpha_{k+4} \tag{20}$$

$$a_k^{(1)} = \alpha_{k+2}^2 - \alpha_{k+1}\alpha_{k+3} \tag{21}$$

$$a_k^{(2)} = \alpha_{k+1}^2 - \alpha_k\alpha_{k+2} \tag{22}$$

$$a_k^{(3)} = \alpha_{k+1}\alpha_{k+4} - \alpha_{k+2}\alpha_{k+3} \tag{23}$$

$$a_k^{(4)} = 2\alpha_{k+1}\alpha_{k+3} - \alpha_{k+2}^2 - \alpha_k\alpha_{k+4} \tag{24}$$

$$a_k^{(5)} = \alpha_k\alpha_{k+3} - \alpha_{k+1}\alpha_{k+2}. \tag{25}$$

Let $\lambda_p = x$ and $\lambda_q = y$ and define functions $f(x, y)$ and $g(x, y)$ from (19) by

$$f(x,y) = a_k^{(0)} + a_k^{(1)}(x+y)^2 + a_k^{(2)}x^2y^2 + a_k^{(3)}(x+y) + a_k^{(4)}xy + a_k^{(5)}xy(x+y), \tag{26}$$

$$g(x,y) = a_{k+1}^{(0)} + a_{k+1}^{(1)}(x+y)^2 + a_{k+1}^{(2)}x^2y^2 + a_{k+1}^{(3)}(x+y) + a_{k+1}^{(4)}xy + a_{k+1}^{(5)}xy(x+y). \tag{27}$$

The solution of the non-linear system $f(x, y) = 0$, $g(x, y) = 0$ will yield approximations to λ_p and λ_q. Thereafter an approximation to λ_r may be obtained from (16) It is obvious that we could solve for one of (λ_1, λ_2), (λ_2, λ_3) or (λ_1, λ_3), from the non-linear system.

LEMMA 2.1. *The only possibility for r in* (16) *is $r = 1$.*

Proof. Since $\alpha_k = \mathcal{O}(\lambda_1^k)$ we let $\alpha_k = K\lambda_1^k$ for some constant K. Thus (16) becomes

$$\lambda_r = \frac{K\lambda_1^{k+3} - K(\lambda_p + \lambda_q)\lambda_1^{k+2} + K\lambda_p\lambda_q\lambda_1^{k+1}}{K\lambda_1^{k+2} - K(\lambda_p + \lambda_q)\lambda_1^{k+1} + K\lambda_p\lambda_q\lambda_1^{k}} = \lambda_1.$$

□

Thus while it is possible to solve for λ_1 from the non-linear system, this is not advocated. Rather we choose to determine the second and third modes from (26)-(27) and then use (16) to determine the first mode. For this we require starting values $\lambda_2^{(0)}$ and $\lambda_3^{(0)}$. We label the solutions of the non-linear system by $\lambda_2^{(1)}$ and $\lambda_3^{(1)}$. Substituting $\lambda_p = \lambda_2^{(1)}$ and $\lambda_q = \lambda_3^{(1)}$ into (16) shall yield $\lambda_r = \lambda_1^{(1)}$. Further refinements of $\lambda_i^{(1)}$ shall be labelled $\lambda_i^{(2)}$, $\lambda_i^{(3)}$, for $i = 1, 2, 3$. The Algorithm below summarizes our technique.

Algorithm.

choose $\mathbf{x}_0 \in \mathbb{R}^n$ randomly, $\|\mathbf{x}_0\|_2 = 1$.
for $i = 1$ to k **do**
  $\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0$
  $\mathbf{x}_0 = \mathbf{x}_1$
end for
for $i = 0$ to 5 **do**
  $\alpha_{k+i} = \langle \mathbf{A}^{k+i}\mathbf{x}_0, \mathbf{x}_0 \rangle$
end for
for $i = 0$ to 5 **do**
  code $a_k^{(i)}$ and $a_{k+1}^{(i)}$ from (20)-(25)
end for
code functions $f(x, y)$ and $g(x, y)$

choose $\lambda_2^{(0)}$ and $\lambda_3^{(0)}$
call non-linear solver **nlsolve** to obtain $\lambda_2^{(1)}$ and $\lambda_3^{(1)}$
evaluate $\lambda_1^{(1)}$ from (17)
for $i = 1$ to 3 **do**
  use $\mathbf{x}_0$ from 1
  for $j = 2$ to 3 **do**
    solve $\left(\mathbf{A} - \lambda_i^{(1)}\mathbf{I}\right)\mathbf{x}_1 = \mathbf{x}_0$ for $\mathbf{x}_1$
    $\mathbf{x}_1 = \frac{\mathbf{x}_1}{\|\mathbf{x}_1\|}$
    $\lambda_i^{(j)} = \langle \mathbf{A}\mathbf{x}_1, \mathbf{x}_1 \rangle$
    $\mathbf{x}_0 = \mathbf{x}_1$
  end for
end for

THEOREM 2.1. *For a symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ with distinct eigenvalues $\lambda_{N-i} \leq \lambda_{N+1-i}, i = 0, 1, \cdots N$, $N \leq n$ and $\mathbf{x} \in \mathbb{R}^n$ with $\|\mathbf{x}\|_2 = 1$, we have*

$$\lambda_N \leq \langle \mathbf{A}\mathbf{x}, \mathbf{x} \rangle \leq \lambda_1. \tag{28}$$

P r o o f. We prove the right hand side of (28), as the left hand side is proved in a similar manner.

$$\begin{aligned} \langle \mathbf{A}\mathbf{x}, \mathbf{x} \rangle &= \left\langle \sum_{i=1}^{N} c_i \lambda_i \mathbf{u}_i, \sum_{j=1}^{N} c_j \mathbf{u}_j \right\rangle \\ &= \sum_{i=1}^{N} c_i^2 \lambda_i \\ &\leq \lambda_1 \sum_{i=1}^{N} c_i^2 \\ &\leq \lambda_1 \langle \mathbf{x}, \mathbf{x} \rangle \\ &\leq \lambda_1. \end{aligned}$$

□

We note that the arrangement of eigenvalues in (1) and Theorem 2.1 agree for positive definite matrices, while for symmetric matrices they may differ. Consider for example the case when λ_N may be negative and the dominant mode. Let $\mathbf{e}_i$ be the standard basis vectors in $\mathbb{R}^n$ and $\mathbf{e} = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \mathbf{e}_i$. With $\mathbf{e}_i$ and $\mathbf{e}$ in theorem 2.1 we have

$$\lambda_N \leq a_{ii} \leq \lambda_1, \tag{29}$$

TABLE 1. Iterates for Example 2.1, $k = 4$, $\star$ - Julia solution

i	$\lambda_3^{(i)}$	$\lambda_2^{(i)}$	$\lambda_1^{(i)}$
0	8.0	16.0	
1	8.487650857031687	10.944130708600321	38.150151999102306
2	8.891929083181118	10.955165340452973	38.150151999442870
3	8.737124078747227	10.955119263606380	38.150151999442876
$\star$	8.736398057639342	10.955119264010982	38.150151999442855

and

$$\lambda_N \leq \frac{1}{n}\sum_{i,j=1}^{n} a_{ij} \leq \lambda_1, \tag{30}$$

respectively. We shall use these two relations to infer weak approximations to $\lambda_2^{(0)}$ and $\lambda_3^{(0)}$.

EXAMPLE 2.1. The positive definite matrix $\mathbf{A}$ given by

$$\mathbf{A} = \begin{bmatrix} 8 & 6 & 4 & 8 & 6 \\ 6 & 16 & 6 & 9 & 9 \\ 4 & 6 & 14 & 4 & 4 \\ 8 & 9 & 4 & 10 & 5 \\ 6 & 9 & 4 & 5 & 16 \end{bmatrix},$$

has $m_i = 1$ for $i = 1, 2, \cdots 5$. From (29) we have $\lambda_5 \leq 8, 16 \leq \lambda_1$ and from (30) we have $37.2 \leq \lambda_1$. We thus choose $\lambda_2^{(0)} = 16$ and $\lambda_3^{(0)} = 8$. Our results are presented in Table 1.

EXAMPLE 2.2. The positive definite matrix $\mathbf{A}$ given by [11]

$$\mathbf{A} = \begin{bmatrix} 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 6 & 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 5 & 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

has $m_i = 1$ for $i = 1, 2, \cdots 7$. Since (30) yields $\lambda_1 \geq 20$, we safely choose $\lambda_2^{(0)} = 7$ and $\lambda_3^{(0)} = 1$. Results are presented in Table 2.

TABLE 2. Iterates for Example 2.2,$K = 4$, $\star$ - Julia solution

i	$\lambda_3^{(i)}$	$\lambda_2^{(i)}$	$\lambda_1^{(i)}$
0	1.0	7.0	
1	0.9793886240731051	2.6178634072214764	22.880782741943560
2	0.9991918942033494	2.6180338192233180	22.880782741943563
3	0.9999973965603144	2.6180339887498910	22.880782741943555
$\star$	1.0	2.6180339887498905	22.880782741943566

EXAMPLE 2.3. The symmetric matrix $\mathbf{A}$ given by [11]

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 0 & 1 & 2 \\ 2 & 4 & 5 & -1 & 0 & 3 \\ 3 & 5 & 5 & -2 & -3 & 0 \\ 0 & -1 & -2 & 1 & 2 & 3 \\ 1 & 0 & -3 & 2 & 4 & 5 \\ 2 & 3 & 0 & 3 & 5 & 6 \end{bmatrix},$$

has $m_1 = m_2 = m_3 = 2$. With $\lambda_3^{(0)} = 1.0$ and $\lambda_2^{(0)} = 6.0$, the non-linear solver gives the results -1.6963154266448344 and 12.427957511551377, while (16) gives 12.411336410744177. We thus realize that the non-linear solver has converged to an approximation to λ_1. Therefore we adjust the starting values to $\lambda_3^{(0)} = 1.0$ and $\lambda_2^{(0)} = 5.0$ and present the results in Table 3. We note that (30) gives $\lambda_1 \geq 10.33$.

TABLE 3. Iterates for Example 2.3, $k = 3$, $\star$ - Julia solution

i	$\lambda_3^{(i)}$	$\lambda_2^{(i)}$	$\lambda_1^{(i)}$
0	1.0	5.0	
1	0.29806558556995827	-1.6963227701119097	12.411336411721404
2	0.28540359025218100	-1.6963228483959059	12.411336411721406
3	0.28498642551671400	-1.6963228483959134	12.411336411721406
$\star$	0.28498643667451135	-1.6963228483959136	12.411336411721406

EXAMPLE 2.4. Here we consider a symmetric pentadiagonal matrix $\mathbf{A}$ of order 10 [11], with the elements defined by $a_{i,i+1} = a_{i+1,i} = 1$, $i = 1, 2, \cdots 9$, $a_{i,i+2} = a_{i+2,i} = 1$, $i = 1, 2, \cdots 8$, $a_{11} = a_{99} = -1$ and $a_{ij} = 0$ otherwise.

The eigenvalues are given by

$$\lambda_i = \left[\frac{1}{2} - 2\cos\left(\frac{k\pi}{11}\right)\right]^2 - \frac{9}{4},$$

TABLE 4. Iterates for Example 2.4, $k = 7$, $\star$ - Julia solution

i	$\lambda_3^{(i)}$	$\lambda_2^{(i)}$	$\lambda_1^{(i)}$
0	-1.0	0	
1	−2.1846834806288060	2.5120004078949414	3.6014922696845706
2	−2.1945327394890652	2.5133454218165980	3.6014930128912184
3	−2.2034648048291210	2.5133370916803606	3.6014930128913570
$\star$	−2.2036156237755650	2.5133370916661337	3.6014930128913583

TABLE 5.

Ex	$\lvert\alpha_k\rvert$	$\lvert e_k\rvert$
1	3.9×10^{12}	28.9
2	6.5×10^{10}	2.3
3	3.4×10^{6}	991.3
4	4.0×10^{7}	1.4

and are clearly distinct. We choose $\lambda_3^{(0)} = 0$ and $\lambda_2^{(0)} = -1$ and present the results in Table 4. We note that (30) gives $\lambda_1 \geq 3.2$.

Table 5 summarizes the absolute values of α_k and e_k for our examples.

3. **Discussion**

Table 1 exhibits very accurate results for λ_1 and λ_2 and slightly less so for λ_3. Table 2 gives excellent results for all three dominant modes. From Example 2.3 we realize that in order to determine all three modes, we require $\lambda_3^{(0)}$ and $\lambda_2^{(0)}$ to be not close to the value of λ_1, otherwise we encounter the risk of converging to λ_1 from nlsolve, a situation we would like to avoid. Our algorithm also succeeds when the eigenvalues are close together or clustered. In this case it is necessary to choose larger values for k. Also if a negative mode is dominant, it is also computed as per Example 2.4. It is obvious that better approximations to $\lambda_3^{(0)}$ and $\lambda_2^{(0)}$, will accelerate the convergence of the non-linear solver. Table 5 shows that $|\frac{e_k}{\alpha_k}| << 1$, justifying out omission of e_k in our computation.

4. **Conclusion**

We have provided a robust algorithm for determining the first three dominant eigenmodes of a real symmetric matrix. Our method succeeds even if the dominant modes have algebraic multiplicities greater than one. It is also possible to determine the second and third mode independent of the first mode. Our examples justify the effectiveness of the technique.

References

[1] T. D. Frank, Eigenvalue analysis of SARS-CoV-2 viral load data: Illustration for eight COVID-19 patients, *Int. J. Data Sci. Anal.* **15**, No 3 (2023), 281–290; doi:10.1007/s41060-022-00319-y.

[2] G. Gu, F. Xie, Ke Zhang, A two-step matrix splitting iteration for computing the PageRank, *Journal of Computational and Applied Mathematics*, **278** (2015), 19–28.

[3] D. A. Gregory, D. Hershkowitz, S. J. Kirkland, The spread of the spectrum of a graph, *Linear Algebra and its Applications*, **332–334** (2001), 23–35.

[4] H. Wolkowicz, G. P. H. Styan, Bounds for eigenvalues using traces, *Linear Algebra and its Applications*, **29** (1980), 471–506.

[5] R. Sharma, R. Kumar, R. Saini, Note on bounds for eigenvalues using traces, *arXiv:1409.0096v1, Functional Analysis* (2014).

[6] P. Singh, V. Singh, S. Singh, New bounds for the maximal eigenvalues of positive definite matrices, *International Journal of Applied Mathematics*, **35**, No 5 (2022), 685–691; doi:10.12732/ijam.v35i5.4.

[7] P. Singh, S. Singh, V. Singh, Results on bounds of the spectrum of positive definite matrices by projections, *Aust. J. Math. Anal. Appl.*, **20**, No 2 (2023), Art. 3, 10 pp.

[8] D. K. Faddeev, V. N. Faddeeva, *Computational Methods of Linear Algebra*, WH Freeman and Company, San Francisco and London (1963).

[9] R. Horn, C. A. Johnson, *Matrix Analysis*, Cambridge University Press (2012), 345–346.

[10] K. V. Singh, Y. M. Ram, Numerical deflation of the transcendental eigenvalue problem, *Mechanical Systems and Signal Processing*, **83** (2017), 522–532.

[11] R. T. Gregory, D. L. Karney, *A Collection of Matrices for Testing Computational Algorithms*, Robert E Krieger Publishing Company, New York (1978).

[12] B. Boehmke, B. M. Greenwell, *Hands-On Machine Learning with R*, Chapman and Hall (2019), 343–396; ISBN 978-1-138-49568-5.

[13] I. T. Jolliffe, J. Cadima, Principal component analysis: A review and recent developments, Philosophical Transactions of the Royal Society A, **374** (2016); doi:10.1098/rsta.20150202.

[14] B. Venkataseshaiah, K. N. Roopadevi, S. Michahial, Image compression using singular value decomposition, *International Journal of Advanced Research in Computer and Communication Engineering*, **5**, No 12 (2016).

[15] M. Manathunga, H. M. Aktulga, A. W. Götz, K. M. Merz, Quantum mechanics/molecular mechanics simulations on NVIDIA and AMD Graphics processing units, *Journal of Chemical Information and Modeling*, **63**, No 3 (2013), 711–717.

[16] M. Zareh, L. Sabattini, C. Secchi, Distributed Laplacian eigenvalue and eigenvector estimation in multi-robot systems, *Distributed Autonomous Robotic Systems. Springer Proceedings in Advanced Robotics*, **6**, (2018), Springer Cham; doi:10.1007/978-3-319-73008-0_14.

[17] M. Slavkovic, D. Jevtic, Face recognition using eigenface approach, *Serbian Journal of Electrical Engineering*, **9**, No 1 (2012), 121–130.

[18] R. Drikvandi, O. Lawal, Sparse principal component analysis for natural language processing, *Annals of Data Science*, **10**, No 1 (2023), 25–41; doi:10.1007/s40745-020-00277-x.

[19] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia (2000).