Numerical Approximation of Bounds of $\mu$-Values for a Family of Pascal Matrices

Mutti-Ur Rehman$^{1,*}$ and Zobia Arif$^2$

$^1$ Department of Mathematics, Sukkur IBA University, Sukkur 65200, Pakistan
$^2$ Department of Mathematics, The University of Lahore, Gujrat campus, Gujrat 50700, Pakistan

Received: 25 Nov. 2018, Revised: 28 Dec. 2018, Accepted: 19 Jan. 2019
Published online: 1 Mar. 2019

Abstract: In this article, we present numerical approximations to lower bounds of Structured Singular Values (SSV) for a family of Pascal matrices. In mathematics, particularly in matrix theory, Pascal matrix is an infinite matrix containing the binomial coefficients as its elements. The obtained lower bounds of SSV are then compared with the well-known MATLAB routine mussv available in MATLAB Control Toolbox.

Keywords: structured singular values, block diagonal perturbation, stability radius, gradient system of ODE's

1 Introduction

Pascal’s triangle is usually represented in the form of a square matrix which possesses two different forms: the very first is lower triangular matrix, say a matrix $P_n$ or even as a full symmetric matrix say $S_n$. Also, it’s obvious that $S_nS_n^T$, where $S_n$ represents a transpose of matrix $S_n$, is nothing but the cholesky decomposition of symmetric matrix $S_n$. While the other side of picture contains a matrix $P_n$ which can be decomposed by a special summation matrices.

In the present article, our main objective is to discuss a numerical method in order to approximate SSV for a family of Pascal matrices. The $\mu$-values introduced by Doyle [8] is a well-known mathematical tool in control which discusses stability and synthesis of the linear-control systems subject to certain class of uncertainties. The perturbation structures addressed by SSV are very generic. These structures allow to cover all kinds of parametric perturbations which can be incorporated into the linear control system by means of both real and complex Linear Fractional Transformations LFT’s. We refer to [3,5,6,7,8] and the references therein for more details.

The computation of an exact value of SSV especially for matrices with higher dimensions appears to be NP-hard [2]. There has been much written work about the approximation of the bounds of SSV. Almost all of numerical methods, which are used in practice, approximate both upper and lower bounds of SSV. The computation of an upper bound of the SSV is to provide sufficient conditions which guarantees the robust stability analysis of feedback systems, while on the other hand a lower bound provides sufficient conditions which guarantees the instability of the linear feedback systems.

The well-known MATLAB function mussv available in the Matlab Control Toolbox approximates an upper bound of SSV by means of the well-known methodologies-the one like diagonal balancing technque and Linear Matrix Inequality technique (LMI) [4]. Furthermore, an approximation of lower bounds of SSV is by means of the generalization of power technique [9].

Definition 1. [1]. The $(n+1) \times (n+1)$ Pascal matrix $P_n$ is defined as

$$P_n(i,j) := \binom{i}{j}; \quad i,j = 0,...,n$$

with $\binom{i}{j} = 0; \quad j > i$.

Further we define the matrices like $I_n, Q_n, D_n$ as;

$$I_n := \text{diag}(1,1,\ldots,1);$$

$$Q_n(i,j) := 1, \quad j \leq i; \quad Q_n(i,j) := 0, \quad j > i;$$

$^*$ Corresponding author e-mail: mutti.rehman@iba-suk.edu.pk
\[ D_n := 1, \ i = 0, \ldots, n; \]
\[ D_n(i+1,i) := -1, \ i = 0, \ldots, n-1; \]
\[ D_n(i,j) := 0, \ j > i \ or \ j < i-1. \]

The Pascal matrix \( P_n \) is characterized by its construction rule, that is,
\[ P_n(i,j) := P_n(i,0) = 1, \ i = 0, \ldots, n; \ P_n(i,j) := 0 \ j > i. \]
\[ P_n(i,j) := P_n(i-1,j) + P_n(i-1,j-1) \ i, \ j = 1, \ldots, n. \]

Let’s consider the \( n \)-dimensional either a real (or a complex) matrix \( M \). The matrix \( M \) could be either square or rectangular, real (or complex) matrix and also consider a family of block diagonal matrices \( \Delta_G \).

**Definition 2.** Suppose that \( M \) is a \( n \)-dimensional either square or a rectangular, real (or complex) matrix and also consider a family of block diagonal matrices \( \Delta_G \). Then the SSV known as \( \mu \)-value is defined as:
\[ \mu_{\Delta_G}(M) := \frac{1}{\min \{ \| \Delta \|_2 : \Delta \in \Delta_G, \det(I - M\Delta) = 0 \} }. \]

We also consider the special case when the set \( \Delta_G \) allows us to have only pure complex uncertainties. We denote \( \Delta_G^c \) instead of \( \Delta_G \) for the family of complex block diagonal matrices. The matrix \( \Delta \in \Delta_G^c \) implies the fact that \( \exp(i\varphi)\Delta \in \Delta_G^c \) for any scalar \( \varphi \in \mathbb{R} \). This lead us with the fact that \( \Delta \in \Delta_G^c \) in such a way that \( \rho(M\Delta) = 1 \) if and only if there exists the perturbation \( \Delta' \in \Delta_G^c \) having the same unit 2-norm such that the matrix \( M\Delta' \) has the maximum eigenvalue exactly equal 1, in turn this implies \( \det(I - M\Delta') = 0 \). The above discussion allows us to write down the following alternative expression for \( \mu \)-value, that is:
\[ \mu_{\Delta_G^c}(M) = \frac{1}{\min \{ \| \Delta \|_2 : \Delta \in \Delta_G^c, \rho(M\Delta) = 1 \} }. \]

Next, we give the definition of a matrix-valued function \( \Delta(t) \) which acts as local extremizer and maximizes the modulus of the greatest eigenvalue \( \lambda_1(t) \).

**Definition 3.** [33]. A matrix-valued function \( \Delta(t) \in \Delta_G^c \) such that \( \| \Delta(t) \|_2 \leq 1 \) and \( eM\Delta \) possesses a greatest eigenvalue \( \lambda_1(t) \) that locally maximizes the modulus of \( \Delta_G^c(M) \) is known as a local extremizer.

### 2 \( \mu \)-value based on structured \( \epsilon \) spectral value sets

The structured epsilon spectral value set of given matrix \( M \in \mathbb{C}^{n \times n} \) with respect to a perturbation level say \( \epsilon \) is given as:
\[ \Lambda_{\epsilon G}^\Delta(M) = \{ \lambda \in \Lambda(eM\Delta) : \Delta \in \Delta_G \}. \]

In Equ. (3), the quantity \( \Lambda(\cdot) \) express the spectrum of a matrix while the admissible perturbation \( \Delta \) possesses a unit 2-norm that is \( \| \Delta \|_2 = 1 \). For the special case when we have purely complex perturbations that is \( \Delta_G^c \), the structured spectral value set \( \Lambda_{\epsilon G}^\Delta(M) \) is nothing but simply a disk having its centered at the origin. While for a more generic case that is: of mixed complex and real uncertainties, the set
\[ \Sigma_{\epsilon G}^\Delta(M) = \{ \xi = 1 - \lambda : \lambda \in \Lambda_{\epsilon G}^\Delta(M) \}. \]

allows us to express \( \mu \)-value as:
\[ \mu_{\Delta_G^c}(M) = \frac{1}{\arg \min \{0 \in \Sigma_{\epsilon G}^\Delta(M) \} }. \]

For a purely complex uncertainties, the underlying set \( \Delta_G^c \) allows us to write down the alternative form of SSV as:
\[ \mu_{\Delta_G^c}(M) = \frac{1}{\arg \min \{ \max |\lambda| = 1 \} }. \]

Here, \( \lambda \in \Lambda_{\epsilon G}^\Delta(M) \).

#### 2.1 The mathematical Problem

We consider the following optimization problem
\[ \xi(\epsilon) = \arg \min |\xi|, \]
where \( \xi \in \Sigma_{\epsilon G}^\Delta(M) \) for a fixed value of perturbation level \( \epsilon \), that is \( \epsilon > 0 \). From above discussion on SSV, \( \mu_{\Delta_G^c}(M) \) is the reciprocal of the minimum value of perturbation level \( \epsilon \) so that \( \xi(\epsilon) = 0 \) which suggests us to give a two-level algorithm, that is, inner and outer algorithm: In the inner algorithm, we solve Equ. (7). While for the outer algorithm, we first vary perturbation level \( \epsilon \) by means of some iterative method which helps to exploits the knowledge of the computation of exact derivative of an extremizer say \( \Delta(\epsilon) \) with respect to perturbation level \( \epsilon \). We solve the optimization problem addressed as in Equ. (6) by first solving a gradient system of Ordinary Differential Equations (ODE’s). This computation only produces a local minimum which, in turn, gives an upper bound for perturbation level \( \epsilon \) and hence as a result one obtains the lower bound for \( \mu_{\Delta_G^c}(M) \). The purely complex uncertainties set \( \Delta_G^c \) can be addressed by taking the inner algorithm to compute a local optima for the maximization problem as addressed below:
\[ \lambda(\epsilon) = \arg \max |\lambda|, \]
where in Equ. (8) \( \lambda \in \Lambda_{\epsilon G}^\Delta(M) \) which then produces a lower bound for \( \mu_{\Delta_G^c}(M) \).

In this section, we establish the solution of the inner problem as mentioned in Eqn. (8). This includes the estimation of the quantity $\mu_{\Delta e}(M)$ for matrix $M \in \mathbb{C}^{n,n}$ while taking into account a pure complex uncertainties set.

$$\Delta e = \{ \text{diag}(\alpha_1, \ldots, \alpha_n, \Delta_1, \ldots, \Delta_F) : \alpha_i \in \mathbb{C}, \Delta_j \in \mathbb{C}^{m_j,m_j} \}.$$  

(9)

In the following Lemma 1, we give the eigenvalue perturbation result in order to approximate the rate of change in the eigenvalue $\lambda(t)$.

**Lemma 1.** Consider a matrix family $\Omega(t) : \mathbb{R} \rightarrow \mathbb{C}^{n,n}$ and consider that $\lambda(t)$ is one of the eigenvalues of $\Omega(t)$ which converges to a simple eigenvalue say $\lambda_1$ of $\Omega_0(0)$ as $t \rightarrow 0$. Then there exists eigenvectors $v_0$ and $w_0$ such that $\lambda(t)$ is analytic near $t = 0$ with

$$\lambda(t)|_{t=0} = \frac{w_0^* \Omega(0)v_0}{w_0^* v_0}.$$  

Here $\Omega = \Omega(0)$ where $\Omega(0)$ denotes the time derivative of matrix $\Omega(t)$ at $t = 0$ and $v_0, w_0$ are right and left eigenvectors of $\Omega_0$ associated to $\lambda_0$, that is, $(\Omega_0 - \lambda_0 I)v_0 = 0$ and $w_0^*(\Omega_0 - \lambda_0 I) = 0$.

Since our goal is to give a solution for the maximization problem as addressed in Eqn. (8). This requires the computation of an uncertainty $\Delta_{\text{local}}$ such that $\mu(\varepsilon \Delta_{\text{local}})$ has the maximum growth among all admissible perturbations $\Delta \in \Delta e$ so that $\Delta$ having a unit 2-norm that is $\|\Delta\|_2 \leq 1$. In the following we call $\lambda_1$ to be the greatest eigenvalue if $|\lambda_1|$ equals the spectral radius of the matrix $\mu(\varepsilon \Delta_{\text{local}})$.

The following theorem provides the characterization of local extremizers $\Delta(t)$.

**Theorem 1.** [11]. The block diagonal matrix

$$\Delta_{\text{local}} = \{ \text{Diag}(\alpha_1, \ldots, \alpha_n, \Delta_1, \ldots, \Delta_F) \},$$

possesses a unit 2-norm, that is, $\|\Delta_{\text{local}}\|_2 = 1$ and acts as a local extremizer of structured epsilon spectral value set $\Lambda e$ when

$$|\alpha_r| = 1 \quad \forall r = 1, \ldots, n \quad \text{and} \quad \|\Delta_h\|_2 = 1 \quad \forall h = 1, \ldots, F.$$  

3.1 A system of ODEs to compute extremal points of $\Lambda e$.

In order to approximate a local maximizer $\Delta(t)$ for greatest eigenvalue $|\lambda_1(t)|$, with $\lambda_1(t) \in \Lambda e$ we first develop a matrix-valued function $\Delta(t)$ that maximizes the greatest eigenvalue $\lambda_1(t)$ of $(\varepsilon M\Delta(t))$ which attains the maximal local growth. Secondly, we derive a gradient system of ordinary differential equations which must satisfy the choice of this initial valued matrix $\Delta(0)$.

3.2 The optimization problem [11].

Consider that $\lambda_1 = |\lambda_1| e^{i\theta}$ be the simple eigenvalue having eigenvectors $v, w$ which are normalized such that

$$\|w\|_2 = \|v\|_2 = 1, \quad w^*v = |w^*v|e^{-i\theta}. \quad (10)$$

As a result of Lemma 1, we get

$$\frac{d}{dt}|\lambda_1|^2 = 2|\lambda_1|Re\left(\frac{u^*\Delta v}{e^{i\theta}w^*v}\right) = 2|\lambda_1|Re(u^*\Delta v), \quad (11)$$

where $u = M^*w$.

By considering $\Delta = \Delta_{\text{local}}$, we compute the direction $\Delta = U$ which locally maximizes the growth of the modulus of greatest eigenvalue $\lambda_1$. From this discussion, we get

$$U = \text{diag}(\alpha_1, \ldots, \alpha_N, \nu_1, \ldots, \nu_F) \quad (12)$$

as a solution of the maximization problem

$$U_* = \arg \max \{Re(u^*Uv)\}$$

subject to $Re(\bar{\nu}_i\nu_i) = 0, \quad i = 1 : N,$

and $Re(\bar{\Delta}_j\nu_j) = 0, \quad j = 1 : F. \quad (13)$

The Lemma 2 gives the solution of the optimization problem as discussed in the Eqn. (12).

**Lemma 2.** [11] We make use of the notation as introduced earlier in the above discussion and $v, u$ being partitioned, a solution $U_*$ of the maximization problem discussed in Eqn. (13) is given as

$$U_* = \text{diag}(\alpha_1, \ldots, \alpha_N, \nu_1, \ldots, \nu_F), \quad (14)$$

with

$$\alpha_i = v_i^*u_i - Re(v_i^*u_i\bar{\nu}_i), \quad i = 1, \ldots, N \quad (15)$$

$$\nu_j = \xi_j (u_{N+j}v_{N+j}^*-Re(\bar{\Delta}_j, u_{N+j}v_{N+j}^*\bar{\Delta}_j)), \quad j = 1, \ldots, (16)$$

Here in the solution $U_*$, the coefficient $\nu_i > 0$ is nothing but the reciprocal of the absolute value of the right-hand side in Eqn. (15) and is other than zero while $\nu_i = 1$ else. Similarly the coefficient $\xi_j > 0$ and is the reciprocal of the Frobenius norm of the matrix obtained on the right hand side in Eqn. (16) and is other than zero while $\xi_j = 1$ else.

The result obtained in Lemma 2 can alternatively be expressed as

$$U_* = D_1 P_{\Delta e}(uv^*) - D_2 \Delta. \quad (17)$$

Here $P_{\Delta e}(\cdot)$ is the orthogonal projection while $D_1, D_2 \in \Delta e$ are diagonal matrices where the orthogonal matrix $D_1$ appears a positive matrix.
4 The gradient system of ODEs

Lemma 2 allows us to consider the following ordinary differential equation on the manifold $\Delta_0^*$:

$$\dot{\Delta} = D_1 P_{\Delta_0}(uv^*) - D_2 \Delta.$$  \hspace{1cm} (18)

Here $v(t)$ is an eigenvector having a unit 2-norm and it is associated to a simple eigenvalue $\hat{\lambda}_1(t)$ of the matrix $\varepsilon M \Delta(t)$ associated with a fixed perturbation $\varepsilon > 0$. The ordinary differential equation (18) represent a gradient system because right-hand side is nothing but the projected gradient of $U \mapsto \text{Re}(u^* U v)$. 

4.1 Choice of initial value matrix $\Delta_0$ and $\varepsilon_0$

In two-level algorithm for approximating $\varepsilon_0$ we make use of the admissible perturbation $\Delta$ obtained for the previous value $\varepsilon_1$ as the initial value matrix for the gradient system of ODEs. While in order to gain the locally maximal growth of $|\lambda_1(t)|$ we choose

$$\Delta_0 = C P_{\Delta_0}(wv^*),$$  \hspace{1cm} (19)

the positive diagonal matrix $C$ is taken in such a way that $\Delta_0 \in \Delta_0^*$. While on the other hand a very natural choice for $\varepsilon$ is given as

$$\varepsilon = \frac{1}{\mu_{\Delta_0}(M)},$$  \hspace{1cm} (20)

here $\hat{\mu}_{\Delta_0}(M)$ is the upper bound of $\mu$-value which is approximated by well-known MATLAB function mussv.

4.2 Outer algorithm

In this section, we approximate the lower bound of SSV, $\mu_{\Delta_0}(M)$ by means of outer algorithm. But we note immediately the fact that the principles remain same as discussed in the previous case, so one can treat the case of purely complex uncertainties in detail and provide a brief discussion on the extension to the case of mixed complex and real uncertainties.

5 Numerical Experimentation

In this section we propose the comparison of the lower bounds of $\mu$-values approximated by mussv and the algorithm [11].

Case-I: 2-dimensional Pascal Matrix $P_2^*$

The following matrix $P_2^*$ is generated by using MATLAB command, that is $P_2 = \text{pascal}(2, 2)$. This returns the lower triangular Cholesky factor (up to the signs of the columns) of the Pascal matrix. It is involutary, that is, it is its own inverse.

$$P_2 = \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix}.$$ 

Consider the perturbation set as:

$$\Delta_3 = \{ \text{diag}(\lambda_1) : \lambda_1 \in \mathbb{C}^{2, 2} \}.$$ 

Using Matlab function mussv, we obtain the perturbation $\hat{\Delta}$ with

$$\hat{\Delta} = \begin{bmatrix} 0.2764 & 0.4472 \\ -0.1708 & -0.2764 \end{bmatrix},$$

while $\|\hat{\Delta}\|_2 = 0.6180$. The upper bound is obtained as $\mu_{\text{upper}}^{\mu_{\text{upper}}} = 1.6180$ while the same lower bound is approximated as $\mu_{\text{lower}}^{\mu_{\text{lower}}} = 1.6180$.

Now, by making use of the algorithm [11], we obtain the perturbation $\varepsilon^* \hat{\Delta}$ with

$$\Delta^* = \begin{bmatrix} 0.4472 & 0.7236 \\ -0.2764 & -0.4472 \end{bmatrix},$$

while $\varepsilon^* = 0.6180$ and $\|\Delta^*\|_2 = 1$. The same lower bound is approximated $\mu_{\text{lower}} = 1.6180$ as the one approximated by MATLAB function mussv.

Case-II: 2-dimensional Pascal Matrix $P_2^*$

The following matrix $P_2^*$ is generated by using MATLAB command, that is $P_2 = \text{pascal}(2, 2)$. This returns a transposed and permuted version of $\text{pascal}(n, 1)$. $P_2^*$ is a cube root of the identity matrix.

$$P_2^* = \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix}.$$ 

Consider the perturbation set as:

$$\Delta_3 = \{ \text{diag}(\lambda_1) : \lambda_1 \in \mathbb{C}^{2, 2} \}.$$ 

Using Matlab function mussv, we obtain the perturbation $\hat{\Delta}$ with

$$\hat{\Delta} = \begin{bmatrix} -0.4472 & 0.2764 \\ -0.2764 & 0.1708 \end{bmatrix},$$

while $\|\hat{\Delta}\|_2 = 0.6180$. For this example, we obtain an upper bound $\mu_{\text{upper}}^{\mu_{\text{upper}}} = 1.6180$ while the same lower bound is approximated as $\mu_{\text{lower}}^{\mu_{\text{lower}}} = 1.600$.

Now by making use of algorithm [11], we obtain the perturbation $\varepsilon^* \hat{\Delta}$ with

$$\Delta^* = \begin{bmatrix} -0.7235 - 0.0092i & 0.4472 - 0.0046i \\ -0.4472 + 0.0046i & 0.2763 - 0.0092i \end{bmatrix},$$

while $\varepsilon^* = 0.6181$ and $\|\Delta^*\|_2 = 1$. The lower bound is approximated as $\mu_{\text{lower}} = 1.6132$ which is much better than the one obtained by mussv function.

© 2019 NSP
Natural Sciences Publishing Corp.
involutary, that is, it is its own inverse. The signs of the columns) of the Pascal matrix. It is involuntary, that is, it is its own inverse.

Consider the perturbation set

$$\Delta_3 = \{\text{diag}(\delta_1 I_1, \delta_2 I_2, \delta_3 I_3) : \delta_1, \delta_2, \delta_3 \in \mathbb{R}\}.$$  

Using Matlab function mussv, we obtain the perturbation $\Delta$ with

$$\hat{\Delta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

while $\|\hat{\Delta}\|_2 = 1$. For this example, we obtain the upper bound $\mu_{\text{upper}}^{PD} = 1$ while the same lower bound is approximated as $\mu_{\text{lower}}^{PD} = 1$.

Now by making use of our algorithm [11], we obtain the perturbation $\varepsilon^* \Delta^*$ with

$$\Delta^* = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

while $\varepsilon^* = 1$ and $\|\Delta^*\|_2 = 1$.

Case-IV: 3-dimensional Pascal matrix $P^*_3$

The following matrix $P^*_3$ is generated by using MATLAB command, that is $P^*_3 = \text{pascal}(3, 2)$. This returns a transposed and permuted version of $\text{pascal}(n, 1)$. $P^*_3$ is a cube root of the identity matrix.

$$P^*_3 = \begin{bmatrix} 1 & 1 & 1 \\ -2 & -1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Consider the perturbation set

$$\Delta_3 = \{\text{diag}(\Delta_1) : \Delta_1 \in \mathbb{C}^{3,3}\}.$$  

Using the Matlab function mussv, we obtain the perturbation $\Delta$ with

$$\hat{\Delta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

while $\|\hat{\Delta}\|_2 = 1$. For this example, we have approximated an upper bound $\mu_{\text{upper}}^{PD} = 2.8059$ while the same lower bound is approximated as $\mu_{\text{lower}}^{PD} = 2.8059$. Now, by making use of our algorithm [11], we obtain the perturbation $\varepsilon^* \Delta^*$ with

$$\Delta^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

and $\|\Delta^*\|_2 = 1$. In this case, lower bound is approximated as $\mu_{\text{lower}} = 2.8059$.

Figure 2 shows the comparison of lower and upper bounds of SSV for the matrix-valued function $P(n, w)$ for the various values of frequency $w$ that is $w = 0, 1, 2, 3, 4$ and $n = 3$ computed by MUSSV and algorithm [11].

Case-V: 6-dimensional Pascal matrix $P_6$

The following matrix $P_6$ is generated by using MATLAB command, that is $P_6 = \text{pascal}(6, 1)$. This returns the lower triangular Cholesky factor (up to the signs of the columns) of the Pascal matrix. It is

$${\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}.$$
Consider the perturbation set

\[ \Delta_B = \{ \text{diag}(\Delta_I) : \Delta_I \in \mathbb{C}^{6,6} \} \]

Using the Matlab function mussv, we obtain the perturbation \( \hat{\Delta} \) with

\[ \hat{\Delta} = \begin{bmatrix}
0 & 0.0001 & 0.0004 & 0.0100 & 0.0022 & 0.0042 \\
-0.0001 & -0.0005 & -0.0017 & -0.0043 & -0.0093 & -0.0180 \\
0.0002 & 0.0009 & 0.0030 & 0.0075 & 0.0162 & 0.0313 \\
-0.0002 & -0.0008 & -0.0026 & -0.0067 & -0.0143 & -0.0277 \\
0.0001 & 0.0004 & 0.0012 & 0.0030 & 0.0064 & 0.0124 \\
0 & -0.0002 & -0.0005 & -0.0012 & -0.0022 & \\
\end{bmatrix} \]

while \( \|\hat{\Delta}\|_2 = 0.0548 \). For this example, we approximate the upper bound \( \mu_{PD}^{upper} = 18.2441 \) while the same lower bound is obtained as \( \mu_{PD}^{lower} = 18.2441 \).

Now, by making use of our algorithm [11], we obtain the perturbation \( \varepsilon^* \Delta^* \) with

\[ \Delta^* = \begin{bmatrix}
0.0004 & 0.0023 & 0.0074 & 0.0186 & 0.0401 & 0.0775 \\
-0.0019 & -0.0098 & -0.0314 & -0.0790 & -0.1701 & -0.3288 \\
0.0032 & 0.0170 & 0.0546 & 0.1373 & 0.2956 & 0.5713 \\
-0.0029 & -0.0150 & -0.0483 & -0.1214 & -0.2614 & -0.5052 \\
0.0013 & 0.0067 & 0.0421 & 0.0544 & 0.1172 & 0.2264 \\
-0.0002 & -0.0012 & -0.0039 & -0.0099 & -0.0212 & -0.0410 \\
\end{bmatrix} \]

while \( \varepsilon^* = 1.0000 \). And \( \|\Delta^*\|_2 = 1 \). In this case the obtained lower bound takes the form as \( \mu_{PD}^{lower} = 18.2441 \).

Case-VI: 6-dimensional Pascal matrix \( P_6^* \).

The following matrix \( P_6^* \) is generated by using the MATLAB command, that is \( P_6^* = \text{pascal}(6,2) \). This returns a transposed and permuted version of \( \text{pascal}(n,1) \). \( P_6^* \) is a cube root of the identity matrix.

\[ P_6^* = \begin{bmatrix}
-1 & -1 & -1 & -1 & -1 & -1 \\
5 & 4 & 3 & 2 & 1 & 0 \\
10 & -6 & -3 & -1 & 0 & 0 \\
5 & 4 & 1 & 0 & 0 & 0 \\
-5 & -1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix} \]

Consider the perturbation set

\[ \Delta_B = \{ \text{diag}(\Delta_I, \Delta_{I2}, \Delta_{I3}) : \Delta_I, \Delta_{I2}, \Delta_{I3} \in \mathbb{R} \} \]

Using the Matlab function mussv, we obtain the perturbation \( \hat{\Delta} \) with

\[ \hat{\Delta} = \begin{bmatrix}
-0.0042 & 0.0180 & -0.0313 & 0.0277 & -0.0124 & 0.0022 \\
-0.0022 & 0.0093 & -0.0162 & 0.0143 & -0.0064 & 0.0012 \\
-0.0010 & 0.0043 & -0.0075 & 0.0067 & -0.0030 & 0.0005 \\
-0.0004 & 0.0017 & -0.0030 & 0.0026 & -0.0012 & 0.0002 \\
-0.0001 & 0.0005 & -0.0009 & 0.0008 & -0.0004 & 0.0001 \\
0 & 0.0001 & 0.0002 & 0.0002 & 0.0001 & 0 \\
\end{bmatrix} \]

while \( \|\hat{\Delta}\|_2 = 0.0548 \). For this example, we obtain the upper bound \( \mu_{PD}^{upper} = 5.3454 \) while the same lower bound is approximated as \( \mu_{PD}^{lower} = 5.3440 \). Now, by making use of our algorithm [11], we obtain the perturbation \( \varepsilon^* \Delta^* \) with \( \varepsilon^* \) and \( \Delta^* \) is given as follows.

\[ \Delta^* = \begin{bmatrix}
-0.0025i & 0.0001i & 0.0008i & 0.002i & -0.002i & -0.0004i \\
-0.0014i & 0.0004i & -0.0001i & 0.0004i & -0.0003i & -0.0002i \\
-0.0007i & 0.0004i & -0.0005i & 0.0005i & -0.0003i & -0.0001i \\
-0.0001i & -0.0004i & 0.0008i & 0.0007i & 0.0003i & -0.0001i \\
0.0002i & -0.0013i & 0.0024i & -0.0021i & 0.0009i & -0.0002i \\
0.0005i & -0.0022i & 0.0042i & -0.0037i & 0.0017i & 0.0003i \\
\end{bmatrix} \]

and \( \|\Delta^*\|_2 = 1.0000 \) and \( \varepsilon^* = 0.0548 \). The obtained lower bound is as \( \mu_{PD}^{lower} = 5.3454 \).

Figure 3 shows the comparison of lower and upper bounds of SSV for the matrix-valued function \( P(n,w) \) for the various values of frequency \( w \) that is \( w = 0, 1, 2, 3, 4, 5 \). and \( n = 6 \) when computed by MUSSV and algorithm [11].

6 Conclusion

In this article we have presented the approximation of \( \mu \)-values for the family of Pascal matrices. Different
experiments have been performed while taking into account the different pascal matrices with various dimensions. The experimental results show how the lower bounds of SSV approximated by mssv function and the one approximated by algorithm [11] are related to each others.

Acknowledgement

The first author acknowledges the financial support by Sukkur IBA University. The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References