Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130215

Numerical Approximation of Bounds of μ-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 abvious that $S_nS_n^t$, where S_n^t 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 μ-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 higer 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 thechnique and Linear Matrix Inequlaity 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*

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

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

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

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

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

* Corresponding author e-mail: mutti.rehman@iba-suk.edu.pk

$$D_n := 1,\ i = 0, \dots, n;$$

$$D_n(i+1,i) := -1,\ i = 0, \dots, n-1;$$

$$D_n(i,j) := 0,\ \ j > i\ or\ j < i-1.$$

The Pascal matrix P_n is characterized by its construction rule, that is,

$$P_n(i,j) := P_n(i,0) = 1,\ i = 0, \dots, n;\ \ P_n(i,j) := 0\ \ j > i.$$

$$P_n(i,j) := P_n(i-1,j) + P_n(i-1,j-1)\ i,j = 1, \dots, n.$$

Let's consider the n-dimensional either a real (or a complex) matrix M. The matrix M could be either square or rectangular in nature. Also consider a family of block diagonal matrices $\Delta_{\mathbb{G}}$.

$$\Delta_{\mathbb{G}} = \{Diag(\Gamma_i, \alpha_j I_j) : \Gamma_i \in \mathbb{C}^{m_i,m_i}(\mathbb{R}^{m_i,m_i}), \alpha_j \in \mathbb{C}(\mathbb{R})\},$$

where I_j is an identity matrix with the dimension j, the same as the dimension of the given matrix M.

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 $\Delta_{\mathbb{G}}$. Then the SSV known as μ-value is defined as:*

$$\mu_{\Delta_{\mathbb{G}}}(M) := \frac{1}{\min\{\|\Delta\|_2 : \Delta \in \Delta_{\mathbb{G}}, \det(I - M\Delta) = 0\}}. \quad (1)$$

We also consider the special case when the set $\Delta_{\mathbb{G}}$ allows us to have only pure complex uncertainties. We denote $\Delta_{\mathbb{G}}^*$ instead of $\Delta_{\mathbb{G}}$ for the family of complex block diagonal matrices. The matrix $\Delta \in \Delta_{\mathbb{G}}^*$ implies the fact that $exp(i\varphi)\Delta \in \Delta_{\mathbb{G}}^*$ for any scalar $\varphi \in \mathbb{R}$. This lead us with the fact that $\Delta \in \Delta_{\mathbb{G}}^*$ in such a way that $\rho(M\Delta) = 1$ if and only if there exists the perturbation $\Delta' \in \Delta_{\mathbb{G}}^*$ 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 μ-value, that is:

$$\mu_{\Delta_{\mathbb{G}}^*}(M) = \frac{1}{\min\{\|\Delta\|_2 : \Delta \in \Delta_{\mathbb{G}}^*,\ \rho(M\Delta) = 1\}}. \quad (2)$$

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. *[11]. A matrix-valued function $\Delta(t) \in \Delta_{\mathbb{G}}^*$ such that $\|\Delta(t)\|_2 \leq 1$ and $\varepsilon M\Delta$ possesses a greatest eigenvalue $\lambda_1(t)$ that locally maximizes the modulus of $\Lambda_\varepsilon^{\Delta_{\mathbb{G}}^*}(M)$ is known as a local extremizer.*

2 μ-value based on structured ε 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 ε is given as:

$$\Lambda_\varepsilon^{\Delta_{\mathbb{G}}}(M) = \{\lambda \in \Lambda(\varepsilon M\Delta) : \Delta \in \Delta_{\mathbb{G}}\}. \quad (3)$$

In Equ. (3), the quantity $\Lambda(\cdot)$ express the spectrum of a matrix while the admissible perturbation Δ possesses a unit 2-norm that is $\|\Delta\|_2 = 1$. For the special case when we have purely complex perturbations that is $\Delta_{\mathbb{G}}^*$, the structured spectral value set $\Lambda_\varepsilon^{\Delta_{\mathbb{G}}}(M)$ 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

$$\Sigma_\varepsilon^{\Delta_{\mathbb{G}}}(M) = \{\xi = 1 - \lambda : \lambda \in \Lambda_\varepsilon^{\Delta_{\mathbb{G}}}(M)\}. \quad (4)$$

allows us to express μ-value as:

$$\mu_{\Delta_{\mathbb{G}}}(M) = \frac{1}{\arg\min\{0 \in \Sigma_\varepsilon^{\Delta_{\mathbb{G}}}(M)\}}. \quad (5)$$

For a purely complex uncertainties, the underlying set $\Delta_{\mathbb{G}}^*$ allows us to write down the alternative form of SSV as:

$$\mu_{\Delta_{\mathbb{G}}^*}(M) = \frac{1}{\arg\min\{\max|\lambda| = 1\}}. \quad (6)$$

Here, $\lambda \in \Lambda_\varepsilon^{\Delta_{\mathbb{G}}^*}(M)$.

2.1 The mathematical Problem

We consider the following optimization problem

$$\xi(\varepsilon) = \arg\min|\xi|, \quad (7)$$

where $\xi \in \Sigma_\varepsilon^{\Delta_{\mathbb{G}}}(M)$ for a fixed value of perturbation level ε, that is $\varepsilon > 0$. From above discussion on SSV, $\mu_{\Delta_{\mathbb{G}}}(M)$ is the reciprocal of the minimum value of perturbation level ε so that $\xi(\varepsilon) = 0$ 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 ε by means of some iterative method which helps to exploits the knowledge of the computation of exact derivative of an extremizer say $\Delta(\varepsilon)$ with respect to perturbation level ε. 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 ε and hence as a result one obtains the lower bound for $\mu_{\Delta_{\mathbb{G}}}(M)$. The purely complex uncertainties set $\Delta_{\mathbb{G}}^*$ can be addressed by taking the inner algorithm to compute a local optima for the maximization problem as addressed below:

$$\lambda(\varepsilon) = \arg\max|\lambda|, \quad (8)$$

where in Equ. (8) $\lambda \in \Lambda_\varepsilon^{\Delta_{\mathbb{G}}^*}(M)$ which then produces a lower bound for $\mu_{\Delta_{\mathbb{G}}^*}(M)$.

3 Pure Complex Perturbations [11]

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

$$\Delta^*_{\mathbb{G}} = \{diag(\alpha_1 I_1, ..., \alpha_n I_n; \Delta_1, ..., \Delta_F) : \alpha_i \in \mathbb{C}, \Delta_j \in \mathbb{C}^{m_j,m_j}\}. \tag{9}$$

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 $\Omega(t) : \mathbb{R} \to \mathbb{C}^{n,n}$ and consider that $\lambda(t)$ is one of the eigenvalue of Ω which converges to a simple eigenvalue say λ_1 of $\Omega_0 = \Omega(0)$ as $t \to 0$. Then there exists eigenvectors v_0 and w_0 such that $\lambda(t)$ is analytic near $t = 0$ with*

$$\dot{\lambda}(t)|_{t=0} = \frac{w_0^* \Omega_1 v_0}{w_0^* v_0}.$$

Here $\Omega_1 = \dot{\Omega}(0)$ where $\dot{\Omega}(0)$ denotes the time derivative of matrix $\Omega(t)$ at $t = 0$ and v_0, w_0 are right and left eigenvectors of Ω_0 associated to λ_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 Equ. (8). This requires the computation of an uncertainty Δ_{local} such that $\rho(\varepsilon A \Delta_{local})$ has the maximum growth among all admissible perturbations $\Delta \in \Delta^*_{\mathbb{G}}$ so that Δ having a unit 2-norm that is $\|\Delta\|_2 \leq 1$. In the following we call λ_1 to be the greatest eigenvalue if $|\lambda_1|$ equals the spectral radius of the matrix $(\varepsilon A \Delta_{local})$.

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

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

$$\Delta_{local} = \{Diag(\alpha_1 I_1, ..., \alpha_n I_n; \Delta_1, ..., \Delta_F)\},$$

possesses a unit 2-norm, that is, $\|\Delta_{local}\|_2 = 1$ and acts as a local extremizer of structured epsilon spectral value set $\Lambda_\varepsilon^{\Delta^*_{\mathbb{G}}}(M)$ when

$$|\alpha_r| = 1 \quad \forall\, r = 1, \dots, n \quad and \quad \|\Delta_h\|_2 = 1 \quad \forall h = 1, \dots, F.$$

3.1 A system of ODEs to compute extremal points of $\Lambda_\varepsilon^{\Delta^*_{\mathbb{B}}}(M)$.

In order to approximate a local maximizer $\Delta(t)$ for greatest eigenvalue $|\lambda_1(t)|$, with $\lambda_1(t) \in \Lambda_\varepsilon^{\Delta_{\mathbb{G}}*}(M)$ 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

$$\|w\|_2 = \|v\|_2 = 1, \qquad w^* v = |w^* v| e^{-i\theta}. \tag{10}$$

As a result of Lemma 1, we get

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

where $u = M^* w$.

By considering $\Delta \in \Delta_{\mathbb{G}}$, we compute the direction $\dot{\Delta} = U$ which locally maximizes the growth of the modulus of greatest eigenvalue λ_1. From this discussion, we get

$$U = diag(\omega_1 I_{r_1}, \dots, \omega_s I_{r_N}, \Omega_1, \dots, \Omega_F) \tag{12}$$

as a solution of the maximization problem

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

$$\text{subject to } Re(\overline{\delta}_i \omega_i) = 0,\ i = 1 : N,$$

$$\text{and } \ Re\langle \Delta_j, \Omega_j \rangle = 0,\ j = 1 : F. \tag{13}$$

The Lemma 2 gives the solution of the optimization problem as discussed in the Equ. (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 Equ. (13) is given as*

$$U_* = diag(\omega_1 I_{r_1}, \dots, \omega_N I_{r_N}, \Omega_1, \dots, \Omega_F), \tag{14}$$

with

$$\omega_i = \nu_i \left(v_i^* u_i - Re\left(v_i^* u_i \bar{s}_i\right) s_i\right), \quad i = 1, \dots, N \tag{15}$$

$$\Omega_j = \zeta_j \left(u_{N+j} v_{N+j}^* - Re\langle \Delta_j, u_{N+j} v_{N+j}^* \rangle \Delta_j\right), \quad j = 1, ..., F \tag{16}$$

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 Equ. (15) and is other than zero while $\nu_i = 1$ else. Similarly the coefficient $\zeta_j > 0$ and is the reciprocal of the Frobenius norm of the matrix obtained on the right hand side in Equ. (16) and is other than zero while $\zeta_j = 1$ else.

The result obtained in Lemma 2 can alternatively be expressed as:

$$U_* = D_1 P_{\Delta^*_{\mathbb{B}}}(uv^*) - D_2 \Delta. \tag{17}$$

Here $P_{\Delta^*_{\mathbb{G}}}(\cdot)$ is the orthogonal projection while $D_1, D_2 \in \Delta^*_{\mathbb{B}}$ 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 $\Delta^*_{\mathbb{G}}$:

$$\dot{\Delta} = D_1 P_{\Delta^*_{\mathbb{G}}}(uv^*) - D_2\Delta. \tag{18}$$

Here $v(t)$ is an eigenvector having a unit 2-norm and it is associated to a simple eigenvalue $\lambda_1(t)$ 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 $U \mapsto Re(u^*Uv)$.

4.1 Choice of initial value matrix Δ_0 and ε_0

In two-level algorithm for approximating ε_0 we make use of the admissible perturbation Δ obtained for the previous value ε_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

$$\Delta_0 = C P_{\Delta_{\mathbb{G}}}(wv^*), \tag{19}$$

the positive diagonal matrix C is taken in such a way that $\Delta_0 \in \Delta_{\mathbb{G}}$.

While on the other hand a very natural choice for ε is given as

$$\varepsilon = \frac{1}{\widehat{\mu}_{\Delta_{\mathbb{G}}}(M)}, \tag{20}$$

here $\widehat{\mu}_{\Delta_{\mathbb{G}}}(M)$ is the upper bound of μ-value which is approximated by well-known MATLAB function mussv.

4.2 Outer algorithm

In this section, we approximate the lower bound of SSV, $\mu_{\Delta_{\mathbb{B}}}(M)$ 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 μ-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 $P_2 = pascal(2,1)$. 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:

$$\Delta_{\mathbb{B}} = \{diag(\Delta_1) : \Delta_1 \in \mathbb{C}^{2,2}\}.$$

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

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

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

Now, by making use of the algorithm [11], we obtain the perturbation $\varepsilon^*\Delta^*$ 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 $\mu_{lower} = 1.6180$ 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 $P_2* = pascal(2,2)$. This returns a transposed and permuted version of $pascal(n,1)$. 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:

$$\Delta_{\mathbb{B}} = \{diag(\Delta_1) : \Delta_1 \in \mathbb{C}^{2,2}\}.$$

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

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

while $\|\widehat{\Delta}\|_2 = 0.6180$. For this example, we obtain an upper bound $\mu_{PD}^{upper} = 1.6180$ while the same lower bound is approximated as $\mu_{PD}^{lower} = 1.600$.
Now by making use of algorithm [11], we obtain the perturbation $\varepsilon^*\Delta^*$ 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 $\mu_{lower} = 1.6132$ which is much better than the one obtained by mussv function.

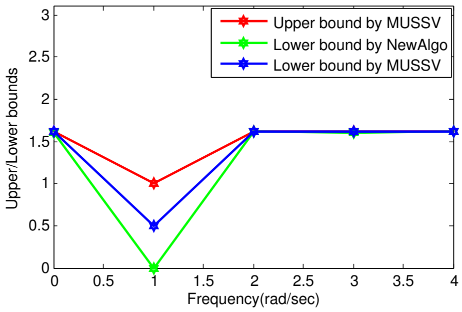

Fig. 1: Comparison of bounds of SSV

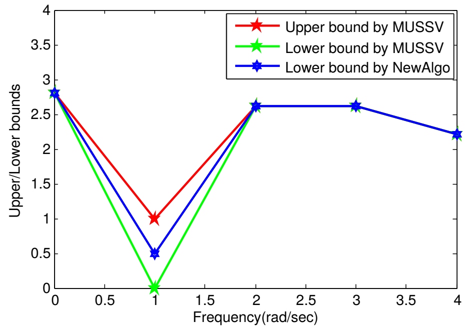

Fig. 2: Comparison of bounds of SSV

Figure 1 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 $n = 2$ when computed by MUSSV and algorithm [11].

Case-III: 3-dimensional Pascal matrix P_3.
The following matrix P_3 is generated by using MATLAB command, that is $P_3 = pascal(3,1)$. 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.

$$P_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -2 & 1 \end{bmatrix}.$$

Consider the perturbation set

$$\Delta_{\mathbb{B}} = \{diag(\delta_1 I_1, \delta_2 I_1, \delta_3 I_1) : \delta_1, \delta_2, \delta_3 \in \mathbb{R}\}.$$

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

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

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

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 $P_3* = pascal(3,2)$. This returns a transposed and permuted version of $pascal(n,1)$. 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

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

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

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

while $\|\widehat{\Delta}\|_2 = 1$. For this example, we have approximated an upper bound $\mu_{PD}^{upper} = 2.8059$ while the same lower bound is approximated as $\mu_{PD}^{lower} = 2.8059$. 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 $\mu_{lower} = 2.8059$.

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 $P_6 = pascal(6,1)$. This returns the lower triangular Cholesky factor (up to the signs of the columns) of the Pascal matrix. It is

involutary.

$$P_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 1 & -3 & 3 & -1 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 \\ 1 & -5 & 10 & -10 & 5 & -1 \end{bmatrix}.$$

Consider the perturbation set

$$\Delta_{\mathbb{B}} = \{diag(\Delta_1) : \Delta_1 \in \mathbb{C}^{6,6}\}.$$

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

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

while $\|\widehat{\Delta}\|_2 = 0.0548$. 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

$$\Delta^* = \begin{bmatrix} 0.0004 & 0.0023 & 0.0074 & 0.0186 & 0.0401 & 0.0775 \\ -0.0019 & -0.0098 & -0.0314 & -0.0790 & -0.1701 & -0.3288 \\ 0.0032 & 0.0170 & 0.0546 & 0.1373 & 0.2956 & 0.5713 \\ -0.0029 & -0.0150 & -0.0483 & -0.1214 & -0.2614 & -0.5052 \\ 0.0013 & 0.0067 & 0.0217 & 0.0544 & 0.1172 & 0.2264 \\ -0.0002 & -0.0012 & -0.0039 & -0.0099 & -0.0212 & -0.0410 \end{bmatrix},$$

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

Case-VI: 6-dimensional Pascal matrix P_6^*.
The following matrix P_6^* is generated by using MATLAB command, that is $P_6* = pascal(6,2)$. This returns a transposed and permuted version of $pascal(n,1)$. P_6^* is a cube root of the identity matrix.

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

Consider the perturbation set

$$\Delta_{\mathbb{B}} = \{diag(\delta_1 I_2, \delta_2 I_2, \delta_3 I_2) : \delta_1, \delta_2, \delta_3 \in \mathbb{R}\}.$$

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

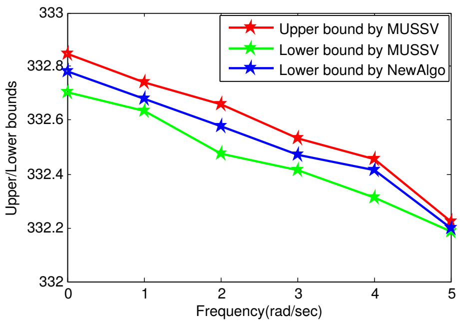

Fig. 3: Comparison of bounds of SSV

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

while $\|\widehat{\Delta}\|_2 = 0.0548$. 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 ε^* and Δ^* is given as follows.

$$\Delta^* = \begin{bmatrix} -0.0781 & 0.3278 & -0.5712 & 0.5064 & -0.2274 & 0.0416 \\ -0.0403 & 0.1693 & -0.2951 & 0.2616 & -0.1175 & 0.0215 \\ -0.0186 & 0.0783 & -0.1364 & 0.1209 & -0.0543 & 0.0099 \\ -0.0073 & 0.0308 & -0.0536 & 0.0476 & -0.0214 & 0.0039 \\ -0.0024 & 0.0099 & -0.0172 & 0.0152 & -0.0068 & 0.0013 \\ -0.0008 & 0.0032 & -0.0055 & 0.0049 & -0.0022 & 0.0004 \end{bmatrix}$$

$$+$$

$$\begin{bmatrix} -0.0025i & 0.0001i & 0.0008i & 0.0002i & -0.0002i & -0.0004i \\ -0.0014i & 0.0004i & -0.0001i & 0.0004i & -0.0003i & -0.0002i \\ -0.0007i & 0.0004i & -0.0005i & 0.0005i & -0.0003i & -0.0001i \\ -0.0001i & -0.0004i & 0.0008i & -0.0007i & 0.0003i & -0.0001i \\ 0.0002i & -0.0013i & 0.0024i & -0.0021i & 0.0009i & -0.0002i \\ 0.0005i & -0.0024i & 0.0042i & -0.0037i & 0.0017i & 0.0003i \end{bmatrix}$$

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

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 μ-values for the family of Pascal matrices. Different experiments have been performed while taking into account the different pascal matrices with varoius dimensions. The experimental results show how the lower bounds of SSV approximated by mussv function and the one approximated by alogorithm [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

[1] Brawer, Robert and P. Magnus, The linear algebra of the Pascal matrix, Linear algebra and its applications, Vol. 174, pp. 13-23 (1992).
[2] Braatz, P. Richard, P. M. Young, J. C. Doyle, and M. Morari, Computational complexity of μ calculation, IEEE Transactions on Automatic Control, Vol. 39, No. 5, pp. 1000-1002 (1994).
[3] Chen, Jie, Michael KH Fan, and Carl N. Nett, Structured singular values with nondiagonal structures. I. Characterizations, IEEE transactions on automatic control, Vol. 41, No. 10, pp. 1507-1511 (1996).
[4] Fan, Michael KH, André L. Tits, and J. C. Doyle, Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics, IEEE Transactions on Automatic Control Vol. 36, No. 1, pp. 25-38 (1991).
[5] D. Hinrichsen and AJ. Pritchard, Mathematical systems theory I: modelling, state space analysis, stability and robustness. Berlin Springer, Vol. 48, pp. xvi+-804 (2005).
[6] Karow, Michael and Kokiopoulou, Effrosyni and D. Kressner, On the computation of structured singular values and pseudospectra, Systems & Control Letters, Vol. 59, pp. 122-129 (2010).
[7] Karow, Michael, daniel Kressner, Daniel and Françoise Tisseur, Structured eigenvalue condition numbers, SIAM Journal on Matrix Analysis and Applications, Vol. 28, No.4, pp. 1052-1068 (2006).
[8] Packard, Andrew and J. Doyle, The complex structured singular value, Automatica Vol. 29,No.1, pp. 71-109 (1993).
[9] Packard, Andy, Michael KH Fan, and J. Doyle, A power method for the structured singular value, Decision and Control, Proceedings of the 27th IEEE Conference on, pp.2132-2137 (1998).
[10] Li, Q., B. Bernhardsson, A. Rantzer, E. J. Davison, P. M. Young, and J. C. Doyle, A Formula for Computation of the Real Stability Radius, Automatica Vol. 31, No. 6, pp. 879-890 (1995).
[11] G. Nicola, R. Mutti-Ur and K. Daniel, A novel iterative method to approximate structured singular values, SIAM Journal on Matrix Analysis and Applications, Vol. 38, No. 2, pp. 361-386 (2017).

Mutti-Ur Rehman received the PhD degree in Applied Mathematics (Numerical Analysis) at SISSA-GSSI Italy and currently working as Assistant professor at Department of Mathematics, Sukkur IBA University, 65200, Sukkur-Pakistan. His research interests are in the areas of applied mathematics.

Zobia Arif is Master (MS) student at The University of Lahore, Gujrat campus, Gujrat-Pakistan. Her research interests are in the areas of applied mathematics.