Numerical approximation of the boundary of numerical range of matrix polynomials

Panayiotis Psarrakos[1] and Charalampos Tsitouras[2]

September 21, 2004

Abstract

The numerical range of an $n \times n$ matrix polynomial $P(\lambda) = A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_1\lambda + A_0$ is defined by

$$W(P) \,=\, \{\lambda \in \mathbb{C} : x^*P(\lambda)x = 0,\, x \in \mathbb{C}^n,\, x^*x = 1\},$$

and plays an important role in the study of matrix polynomials. In this paper, we describe a methodology for the illustration of its boundary, $\partial W(P)$, using recent theoretical results on numerical ranges and algebraic curves.

Keywords: matrix polynomial; eigenvalue; numerical range; boundary; discriminant

AMS Subject Classifications: 15A60; 65D18; 65F30

1 Introduction and preliminaries

Consider a *matrix polynomial*

$$P(\lambda) \,=\, A_m\lambda^m + A_{m-1}\lambda^{m-1} + \cdots + A_1\lambda + A_0, \tag{1}$$

where $A_j \in \mathbb{C}^{n\times n}$ $(j = 0, 1, \ldots, m)$ and λ is a complex variable. The spectral analysis of matrix polynomials leads to the solutions of higher order linear systems of differential equations (or difference equations) with constant coefficients. The suggested references are [3, 8].

A scalar $\lambda_0 \in \mathbb{C}$ is said to be an *eigenvalue* of $P(\lambda)$ in (1) if the system $P(\lambda_0)x = 0$ has a nonzero solution $x_0 \in \mathbb{C}^n$. This solution x_0 is known as an *eigenvector* of $P(\lambda)$ corresponding to λ_0, and the set of all eigenvalues of $P(\lambda)$ is the *spectrum* of $P(\lambda)$, namely, $\sigma(P) = \{\lambda \in \mathbb{C} : \det P(\lambda) = 0\}$.

[1] Department of Mathematics, National Technical University, Zografou Campus, 15780 Athens, Greece (E-mail: ppsarr@math.ntua.gr).

[2] Department of Applied Sciences, TEI of Chalkis, GR34400, Psahna, Greece (E-mail: tsitoura@teihal.gr).

The *numerical range* of $P(\lambda)$ in (1) is defined and denoted by

$$W(P) = \{\lambda \in \mathbb{C} : x^*P(\lambda)x = 0,\ x \in \mathbb{C}^n,\ x^*x = 1\}. \tag{2}$$

Evidently, $W(P)$ is always closed and contains the spectrum of $P(\lambda)$. For the linear pencil $P(\lambda) = I\lambda - A$, $W(P)$ coincides with the *numerical range* (also known as *field of values*) of the matrix A, $F(A) = \{x^*Ax : x \in \mathbb{C}^n,\ x^*x = 1\}$. The suggested reference on $F(A)$ and its properties is [2].

The last decade, the numerical range $W(P)$ in (2) has attracted attention, and several results have been obtained (see e.g., [1, 4, 5, 6, 7, 9, 10, 11, 12, 13, 16]). These results are helpful in investigating and understanding matrix polynomials, and lead to interesting applications of the numerical range on the spectral analysis, the factorization and the stability of matrix polynomials, [4, 11, 16]. To facilitate the presentation, some basic geometrical properties of $W(P)$ are given below, [6, 7, 8, 9, 10].

(A) The numerical range $W(P)$ is not always connected, and it is bounded if and only if $0 \notin F(A_m)$. In this case, $W(P)$ has no more than m (bounded) connected components.

(B) Suppose $\mathcal{G}$ is a bounded connected component of $W(P)$ and for a unit vector $x \in \mathbb{C}^n$, the scalar polynomial $x^*P(\lambda)x$ has exactly $c(\mathcal{G})$ roots in $\mathcal{G}$, counting multiplicities. Then the number $c(\mathcal{G})$ does not depend on x, and $P(\lambda)$ has exactly $n\,c(\mathcal{G})$ eigenvalues in $\mathcal{G}$, counting multiplicities.

(C) For any $\mu \in \partial W(P)$, the origin is a boundary point of $F(P(\mu))$.

(D) If μ is a corner of $W(P)$ and there is a unit $x_\mu \in \mathbb{C}^n$ such that μ is a simple root of the polynomial $x_\mu^*P(\mu)x_\mu$, then μ is an eigenvalue of $P(\lambda)$.

A straightforward procedure for the estimation of $W(P)$ (based on the definition) would be to plot the roots of the polynomial $x^*P(\lambda)x$ for lots and randomly chosen unit vectors $x \in \mathbb{C}^n$. But that would be too costly, and it would probably not accurately depict the boundary of $W(P)$. Algorithms for plotting the boundaries of the numerical ranges of $A\lambda^{2m_1+m_2} + B\lambda^{m_1+m_2} + C\lambda^{m_2}$ and $A\lambda^{m_1+m_2} + (B + \mathrm{i}C)^{m_2}$ $(m_1, m_2 \geq 0)$ for hermitian A, B, C can be found in [5, 13, 14]. An inclusion-exclusion methodology for the estimation of $W(P)$ when $A_m = I$, i.e., for the monic case, is described in [15]. The numerical approximation of $W(P)$ is still an open and challenging problem.

In this paper, a procedure for the approximation of the boundary of $W(P)$ is proposed. This method is the first method for the estimation of the numerical range of a general matrix polynomial besides the application of the definition. The algorithm described in the next section is based on a recent theoretical result of Chien, Nakazato and Psarrakos, [1], which yields an algebraic curve of degree at most $2n(n-1)m$ that contains $\partial W(P)$. Illustrative examples are given in Section 3.

2 The algorithm

For our approach, it is necessary to recall an algebraic criterion for a scalar polynomial to have a multiple root.

Consider a polynomial $p(t) = \alpha_l t^l + \alpha_{l-1} t^{l-1} + \cdots + \alpha_1 t + \alpha_0$ and its derivative $p'(t) = l\alpha_l t^{l-1} + (l-1)\alpha_{l-1} t^{l-2} + \cdots + \alpha_1$, where $\alpha_0, \alpha_1, \ldots, \alpha_l \in \mathbb{C}$ and t is a complex variable. We define the *resultant (Sylvester determinant)* of $p(t)$ and $p'(t)$, that is, the $(2l-1) \times (2l-1)$ determinant

$$\tilde{\mathcal{D}}_p = \begin{vmatrix} \alpha_l & \alpha_{l-1} & \ldots & \ldots & \alpha_0 & 0 & \ldots & 0 \\ 0 & \alpha_l & \alpha_{l-1} & \ldots & \ldots & \alpha_0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & \ldots & 0 & \alpha_l & \alpha_{l-1} & \ldots & \alpha_0 \\ l\alpha_l & (l-1)\alpha_{l-1} & \ldots & \alpha_1 & 0 & 0 & \ldots & 0 \\ 0 & l\alpha_l & (l-1)\alpha_{l-1} & \ldots & \alpha_1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & \ldots & \ldots & 0 & l\alpha_l & \ldots & \alpha_1 \end{vmatrix}.$$

The polynomial $\tilde{D}_p$ is homogeneous in variables $\alpha_0, \alpha_1, \ldots, \alpha_l$ of degree at most $2l-1$, and contains a factor α_l. If $\alpha_l \neq 0$, then the ratio $D_p = \tilde{D}_p / \alpha_l$ is said to be the *discriminant* of $p(t)$, and it is a homogeneous polynomial in $\alpha_0, \alpha_1, \ldots, \alpha_l$ of degree at most $2l-2$. Furthermore, $D_p = 0$ if and only if $p(t)$ has a multiple root, [17]. If $\alpha_l = 0$, then D_p is assumed to be zero.

Let now $P(\lambda) = A_m \lambda^m + A_{m-1} \lambda^{m-1} + \cdots + A_1 \lambda + A_0$ be an $n \times n$ matrix polynomial as in (1) with numerical range $W(P)$ as in (2). In the following result, we formulate the point equation of the curve $\partial W(P)$, [1].

Theorem 1 *The boundary of the numerical range $W(P)$ lies on the algebraic curve*

$$\mathcal{C}(P) = \{u + \mathrm{i}\, v \in \mathbb{C} : u, v \in \mathbb{R},\ D_P(u, v) = 0\}, \tag{3}$$

where $D_P(u, v)$ is the discriminant of the polynomial

$$G_P(t; u, v) = \det(P(u + \mathrm{i}\, v) + t\,[P(u + \mathrm{i}\, v)]^*) \tag{4}$$

with respect to variable t. This discriminant is a polynomial (not necessarily with real coefficients) in $u, v \in \mathbb{R}$ of total degree at most $2n(n-1)m$.

One can see that the polynomial $G_P(t; u, v)$ in (4) satisfies

$$t^n \overline{G_P(\overline{t}^{-1}; u, v)} = \det(t\,[P(u + \mathrm{i}\, v)]^* + P(u + \mathrm{i}\, v)) = G_P(t; u, v).$$

Hence, t_0 is a nonzero root of $G_P(t; u, v)$ with multiplicity k if and only if $\overline{t}_0^{-1}$ is a root of $G_P(t; u, v)$ with the same multiplicity. As a consequence, the modules of the product of all nonzero roots of $G_P(t; u, v)$ is always equal to 1.

We also remark that if $u_0 + \mathrm{i}\, v_0$ $(u_0, v_0 \in \mathbb{R})$ is an isolated point of $W(P)$, then by [9, Theorem 2.1], $P(u_0 + \mathrm{i}\, v_0) = 0$. Consequently, the polynomial $G_P(t; u_0, v_0)$ is identically zero, and by the above discussion, the discriminant $D_P(u_0, v_0)$ is assumed to be zero.

An $n \times n$ matrix polynomial $P(\lambda)$ is called *selfadjoint* if all its coefficients are hermitian matrices. In this case, it is easy to see that $W(P)$ is symmetric with respect to the real axis. Moreover, for every $u \in \mathbb{R}$, the polynomial $G_P(t; u, 0) = \det(P(u) + t\,[P(u)]^*) = (1-t)^n \det P(u)$ has zero discriminant. Thus, if $P(\lambda)$ is selfadjoint, then the algebraic curve $\mathcal{C}(P)$ in (3) contains the real axis (see Example 2 below).

Suppose $W(P) \neq \mathbb{C}$ and $\Omega = [u_{\min}, u_{\max}] \times [\mathrm{i}\, v_{\min}, \mathrm{i}\, v_{\max}]$ (for real $u_{\min}$, $u_{\max}$, $v_{\min}$ and $v_{\max}$) is a given rectangle in the complex plane that contains $W(P)$ or a part of it. Then we can estimate $\partial W(P) \cap \Omega$ by applying the following grid-based algorithm.

Algorithm

Input: The coefficients $A_0, A_1, \ldots, A_m$ of the matrix polynomial $P(\lambda)$, the real bounds $u_{\max}$, $u_{\min}$, $v_{\max}$ and $v_{\min}$, the positive integers N_u and N_v.

Step I Construct the $N_u \times N_v$ grid

$$\{u + \mathrm{i}\, v = u_{\min} + \zeta\, h_u + \mathrm{i}\,(v_{\min} + \xi\, h_v) : \zeta = 0, 1, \ldots, N_u,\ \xi = 0, 1, \ldots, N_v\}$$

of the rectangle $\Omega = [u_{\min}, u_{\max}] \times [\mathrm{i}\, v_{\min}, \mathrm{i}\, v_{\max}]$, where $h_u = (u_{\max} - u_{\min}) \,/\, N_u$ and $h_v = (v_{\max} - v_{\min}) \,/\, N_v$.

Step II For every grid point $u + \mathrm{i}\, v$, repeat:

(a) Construct the scalar polynomial

$$G_P(t; u, v) \;=\; \det(P(u + \mathrm{i}\, v) + t\,[P(u + \mathrm{i}\, v)]^*)$$

in variable t.

(b) Compute the determinant $\tilde{D}_P(u, v)$ and the discriminant $D_P(u, v)$.

Output: The plot of the curve $\mathcal{C}(P) \cap \Omega$.

An implementation of the above algorithm in MATLAB is given in Appendix, where the command `temp/vander(1:L+1)'` estimates the polynomial $G_P(t; u, v)$ in a least square sense and the command `contour` sketches the curve $\mathcal{C}(P) \cap \Omega$. As might be expected, the use of these commands leads to some difficulties near points $u + \mathrm{i}\, v$ of $\partial W(P)$ where the discriminant $D_P(u, v)$ vanishes but has a constant sign in a neighborhood of $u + \mathrm{i}\, v$ (this is the case in Example 2 below), or when parts of $W(P)$ have no interior. Since the mapping $P(\lambda) \mapsto W(P) \cap \Omega$ is continuous with respect to the Hausdorff metric,

this problem can be canceled by perturbing appropriately the original matrix polynomial and/or choosing a more suitable grid. In spite of this weakness, the authors' experiments suggest that the algorithm is remarkably robust.

An important feature of this technique is that it does not depend strongly on the degree m of $P(\lambda)$, which appears only in the computation of the (fixed) matrix $P(u + \mathrm{i}\, v)$ in Step II (a). On the other hand, the cost of the algorithm depends mainly on the size n of $P(\lambda)$ since for every grid point $u + \mathrm{i}\, v$, the calculation of the $(2n-1) \times (2n-1)$ determinant $\tilde{D}_P(u, v)$ is required.

Our methodology also requires *a priori* knowledge of the size and the location of $W(P)$. Thus, in our experiments, before applying the algorithm, we plot the roots of some polynomials of the form $x^* P(\lambda) x$ for unit $x \in \mathbb{C}^n$. In this way, we obtain an approach of a rectangle of interest $\Omega = [u_{\min}, u_{\max}] \times [\mathrm{i}\, v_{\min}, \mathrm{i}\, v_{\max}]$ that contains $W(P)$ or a part of it.

3 Numerical examples

In this section, we present some examples to illustrate the use of the proposed algorithm. The computations were performed in MATLAB.

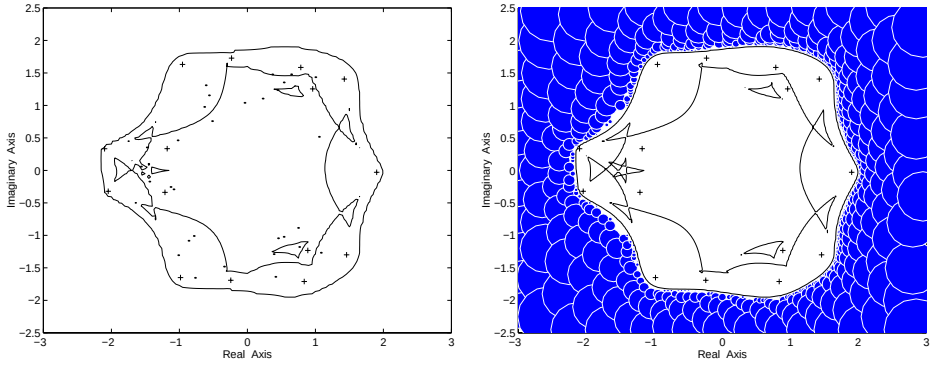

Figure 1: A connected numerical range.

Example 1 For the 5×5 monic matrix polynomial

$$P_1(\lambda) \quad = \quad I\lambda^3 + \frac{1}{10} \begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{i} \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix} \lambda^2$$

Table 1: Efficiency comparisons for $W(P_1)$.

New Method		Method of [15]	
13.6s	$4.5{\cdot}10^{-2}$	16.5s	$1.2{\cdot}10^{-1}$
23.5s	$3.8{\cdot}10^{-2}$	27.0s	$4.2{\cdot}10^{-2}$
50.4s	$6.1{\cdot}10^{-5}$	66.7s	$1.9{\cdot}10^{-2}$
199.0s	$4.2{\cdot}10^{-5}$	364.5s	$2.9{\cdot}10^{-3}$

$$+\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 0 \end{bmatrix}\lambda+\begin{bmatrix} 5 & 4 & 6 & -7 & 0 \\ 0 & 3 & 4 & -2 & 1 \\ 3 & -4 & 0 & 0 & 0 \\ \mathrm{i} & -1 & 0 & 0 & 8 \\ 4 & 3 & -5 & 2 & 0 \end{bmatrix},$$

the roots of a few thousand randomly chosen polynomials of the form $x^*P_1(\lambda)x$ ($x \in \mathbb{C}^5$, $x^*x = 1$) are plotted in the left part of Figure 1. These points do not give a clear picture of $\partial W(P_1)$, and eigenvalues of $P_1(\lambda)$ (marked with '+') appear to lie out of $W(P_1)$. Moreover, it seems that $W(P_1)$ lies in the rectangle $\Omega_1 = [-3,3]\times[-\mathrm{i}\,2.5,\mathrm{i}\,2.5]$. In the right part of the figure, we estimate $\Omega_1 \setminus W(P_1)$ by using the method described in [15], which is valid only for monic matrix polynomials. Our algorithm is applied on a 200×200 grid for drawing the curve $\mathcal{C}(P_1) \cap \Omega_1$ in both parts of the figure. Apparently, the outer branch of the curve is a satisfactory estimation of $\partial W(P_1)$. $\square$

The question of comparison of the new method applied to monic matrix polynomials along with the method proposed in [15] arises in a natural way. Hence, in Table 1, we give for several grids, the execution time together with the distance of the boundary derived by both methods from the most left real point of $\partial W(P_1)$, -2.150078431492111. Furthermore, it is easy to see that the numerical range of the matrix polynomial

$$R(\lambda) \;=\; I\lambda^3 - \begin{bmatrix} 0 & 16 \\ 0 & 0 \end{bmatrix}$$

is the circular disk with centre at the origin and radius equal to 2. In Table 2, we record the execution time and the Hausdorff distance of the computed boundaries from the circle $\partial W(R)$. The results in both tables clearly demonstrate that the new algorithm returns a better approximation of the boundaries at a much smaller cost than the inclusion-exclusion algorithm of [15].

Example 2 The boundary of the numerical range of the selfadjoint quadratic matrix polynomial

$$P_2(\lambda) \;=\; I\lambda^2 + \begin{bmatrix} 0 & -\mathrm{i}\,0.2 & \mathrm{i}\,0.2 \\ \mathrm{i}\,0.2 & 0 & -\mathrm{i}\,0.2 \\ -\mathrm{i}\,0.2 & \mathrm{i}\,0.2 & 0 \end{bmatrix}\lambda + \begin{bmatrix} 0.3 & 0.2 & 0.5 \\ 0.2 & 0.5 & 0.3 \\ 0.5 & 0.3 & 0.2 \end{bmatrix}$$

is accurately plotted in the left part of Figure 2, using an algorithm described in [5] (valid only for monic selfadjoint quadratic matrix polynomials). Choosing a

Table 2: Efficiency comparisons for $W(R)$.

New Method		Method of [15]	
6.7s	$2.9{\cdot}10^{-3}$	9.8s	$2.8{\cdot}10^{-1}$
10.6s	$1.7{\cdot}10^{-3}$	15.2s	$8.4{\cdot}10^{-2}$
24.1s	$7.6{\cdot}10^{-4}$	26.9s	$5.1{\cdot}10^{-2}$
91.5s	$1.9{\cdot}10^{-4}$	128.2s	$3.6{\cdot}10^{-2}$

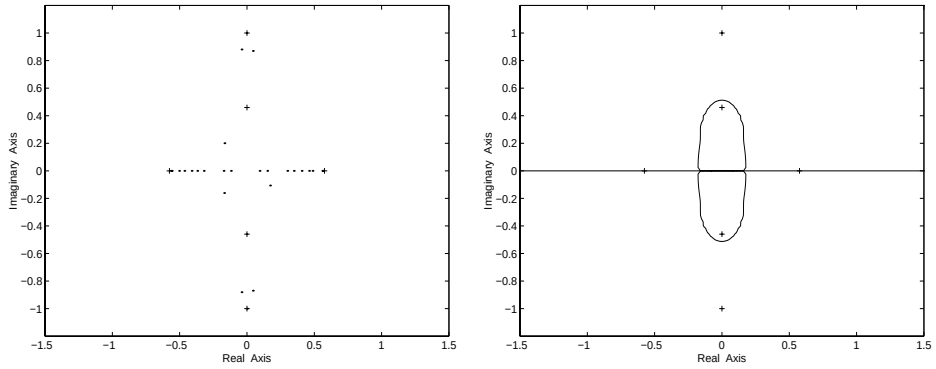

Figure 2: The numerical range $W(P_2)$ and the curve $\mathcal{C}(P_2) \cap \Omega_2$.

100×100 grid of the rectangle $\Omega_2 = [-1, 1] \times [-\mathrm{i}\,1.2, \mathrm{i}\,1.2]$, we draw the curve $\mathcal{C}(P_2) \cap \Omega_2$ in the right part of the figure. The eigenvalues of $P_2(\lambda)$ are also marked with '+'. Clearly, the illustration of $\mathcal{C}(P_2) \cap \Omega_2$ is not complete. At the eigenvalues $u + \mathrm{i}\,v = \pm\mathrm{i}$ of $P_2(\lambda)$ (corners of $W(P_2)$), the scalar polynomial $G_{P_2}(t; u, v)$ (in variable t) is identically zero and the discriminant $D_{P_2}(u, v)$ does not change sign in neighborhoods of these points. As a consequence, the command `contour` cannot plot the nonreal part of $\partial W(P_2)$. Let now $\hat{P}_2(\lambda)$ be the matrix polynomial that follows from $P_2(\lambda)$ by replacing the (3,3)-entry of the constant term of $P_2(\lambda)$, 0.2, with 0.4. Using a 200×200 grid of Ω_2, we estimate the curve $\mathcal{C}(\hat{P}_2) \cap \Omega_2$ in the left part of Figure 3. A neighborhood of the eigenvalue $\lambda = \mathrm{i}$ is magnified in the right part of the figure. The points $\pm\mathrm{i}$ are no corners of $W(\hat{P}_2)$, and the problem is now canceled. $\square$

The new methodology is the only currently known to the authors method for the approximation of unbounded numerical ranges of matrix polynomials (besides the definition). In our last example, we consider this case.

Example 3 The numerical range of the matrix polynomial

$$P_3(\lambda) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \lambda^2 + \begin{bmatrix} 3 & 2 & 0 & 0 \\ -2 & 3 & 2 & 0 \\ 0 & -2 & 3 & 2 \\ 0 & 0 & -2 & 3 \end{bmatrix} \lambda + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

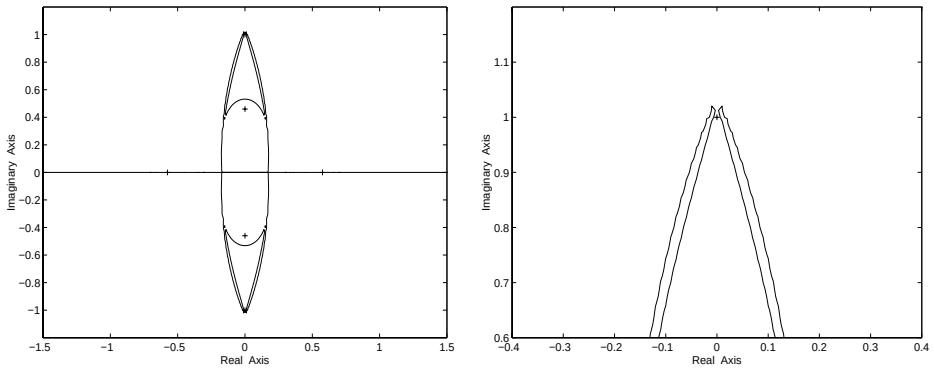

Figure 3: The curve $\mathcal{C}(\hat{P}_2) \cap \Omega_2$.

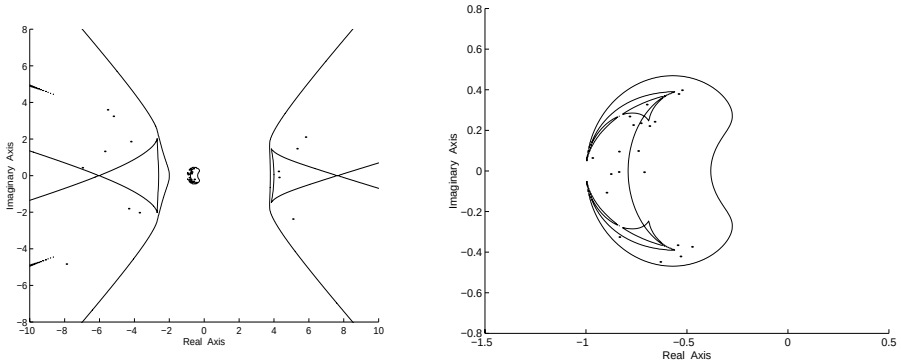

Figure 4: An unbounded numerical range.

is unbounded since the numerical range of the leading coefficient, that is, the interval $[-1, 1]$, contains the origin. The roots of a few thousand randomly chosen polynomials of the form $x^* P_3(\lambda) x$ ($x \in \mathbb{C}^4$, $x^* x = 1$) are plotted in the left part of Figure 4. With these points, we cannot have a clear picture of $W(P_3)$, but $W(P_3)$ appears to have one bounded and two unbounded connected components, confirming Theorem 2.2 in [7]. Choosing the rectangle $\Omega_3 = [-10, 10] \times [-\mathrm{i}\,8, \mathrm{i}\,8]$ and a 400×400 grid, we sketch the curve $\mathcal{C}(P_3) \cap \Omega_3$ in the same part of the figure. The bounded connected component of $W(P_3)$ is magnified in the right part of Figure 4. $\square$

4 Appendix

The MATLAB function implementing the new method follows.

```
function c = nrcurve3(Q,xrange,yrange,h);
% Numerical range of the matrix polynomial
%     n         n-1
%  A x  +  A   x   +  ...  +  A x  +  A
%   n       n-1                1       0
%
% Input: Q = {An;An-1;...;A1;A0}
%        xrange = [left bound on x-axis, right bound on x-axis]
%        yrange = [lower bound on y-axis, upper bound on y-axis]
%        h = length of the grid
%
% Output: c = contour plot points

n1=round((xrange(2)-xrange(1))/h);
h1=(xrange(2)-xrange(1))/n1;
x=xrange(1):h1:xrange(2);
n2=round((yrange(2)-yrange(1))/h);
h2=(yrange(2)-yrange(1))/n2;
y=yrange(1):h2:yrange(2);
d=zeros(n1+1,n2+1);
K=length(Q);L=length(Q{1});i=sqrt(-1);temp=zeros(1,L);
for i1=1:n1+1,
  for i2=1:n2+1,
    z=x(i1)+i*y(i2);

    for t=1:L+1,
      temp(t)=dett(K,t,Q,z);
    end;

    a=temp/vander(1:L+1)';
    dg=zeros(2*L-1,2*L-1);

    for inde=1:L,
      dg(inde,inde:inde+L-1)=(L:-1:1).*(a(L+1:-1:2));
    end;

    for inde=L+1:2*L-1,
      dg(inde,inde-L:inde)=a(L+1:-1:1);
    end;

    d(i1,i2)=det(dg)/dg(L+1,1);
  end;
end;
c=contour(x,y,d',[0 0],'-k');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function u=dett(K,t,Q,z);
```

```
u=Q{1};zz=z;
for i1=2:K,
  u=u+Q{i1}*zz;
  zz=zz*z;
end;
u=det(u+t*u');
```

References

[1] M.-T. Chien, H. Nakazato and P. Psarrakos, Point equation of the boundary of the numerical range of a matrix polynomial, *Linear Algebra Appl.* 347 (2002) 205-217.

[2] R.A. Horn and C.R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, 1991.

[3] I. Gohberg, P. Lancaster and L. Rodman, *Matrix Polynomials*, Academic Press, New York, 1982.

[4] P. Lancaster and P. Psarrakos, Normal and seminormal eigenvalues of matrix functions, *Integral Equations and Operator Theory* 41 (2001) 331-342.

[5] P. Lancaster and P. Psarrakos, The numerical range of selfadjoint quadratic matrix polynomials, *SIAM J. Matrix Anal. Appl.* 23 (2001) 615-631.

[6] H. Langer, A. Markus and C. Tretter, Corners of numerical ranges, *Operator Theory : Advances and Applications* 124 (2001) 385-400.

[7] C.-K. Li and L. Rodman, Numerical range of matrix polynomials, *SIAM J. Matrix Anal. Appl.* 15 (1994) 1256-1265.

[8] A.S. Markus, *Introduction to the Spectral Theory of Polynomial Operator Pencils*, Amer. Math. Society, Providence, Translations of Math. Monographs, Vol. 71, 1988.

[9] J. Maroulas and P. Psarrakos, The boundary of numerical range of matrix polynomials, *Linear Algebra Appl.* 267 (1997) 101-111.

[10] J. Maroulas and P. Psarrakos, On the connectedness of numerical range of matrix polynomials, *Linear Algebra Appl.* 280 (1998) 97-108.

[11] J. Maroulas and P. Psarrakos, On factorization of matrix polynomials, *Linear Algebra Appl.* 304 (2000) 131-139.

[12] H. Nakazato and P. Psarrakos, On the shape of numerical range of matrix polynomials, *Linear Algebra Appl.* 338 (2001) 105-123.

[13] P. Psarrakos, Numerical range of linear pencils, *Linear Algebra Appl.* 317 (2000) 127-141.

[14] P. Psarrakos, Definite triples of Hermitian matrices and matrix polynomials, *J. Comput. Appl. Math.* 151 (2003) 39-58.

[15] P. Psarrakos, On the estimation of the q-numerical range of monic matrix polynomials, *Electronic Transactions on Numerical Analysis* 17 (2004) 1-10.

[16] P. Psarrakos and M. Tsatsomeros, On the stability radius of matrix polynomials, *Linear and Multilinear Algebra* 50 (2002) 151-165.

[17] R. Walker, *Algebraic Curves*, Dover Publ., New York, 1950.