Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

A METHOD FOR THE INVERSE NUMERICAL RANGE PROBLEM*

CHRISTOS CHORIANOPOULOS†, PANAYIOTIS PSARRAKOS†, AND FRANK UHLIG‡

Abstract. For a given complex square matrix A, we develop, implement and test a fast geometric algorithm to find a unit vector that generates a given point in the complex plane if this point lies inside the numerical range of A, and if not, the method determines that the given point lies outside the numerical range.

Key words. Numerical range, Inverse problem, Field of values, Geometric computing, Generating vector.

AMS subject classifications. 15A29, 15A60, 15A24, 65F30, 65J22.

1. Introduction and preliminaries. The *numerical range* (also known as the *field of values*) of a square matrix $A \in \mathbb{C}^{n\times n}$ is the compact and convex set

$$F(A) = \{x^* A\, x \in \mathbb{C} : x \in \mathbb{C}^n,\, x^* x = 1\} \subset \mathbb{C}.$$

The compactness follows readily from the fact that $F(A)$ is the image of the compact unit sphere of $\mathbb{C}^n$ under the continuous mapping $x \longmapsto x^* A\, x$, and the convexity of $F(A)$ is due to Toeplitz [12] and Hausdorff [8]. The concept of the numerical range and related notions has been studied extensively for many decades. It is quite useful in studying and understanding matrices and operators [7, 10], and has many applications in numerical analysis, differential equations, systems theory etc (see e.g. [1, 4, 5, 6]).

Our proposed algorithm to solve the inverse numerical range problem relies only on the most elementary properties of $F(A)$ other than its convexity. Namely, for any matrix $A \in \mathbb{C}^{n\times n}$ and any $\alpha, \beta \in \mathbb{C}$, $F(\alpha A + \beta I_n) = \alpha F(A) + \beta$ where I_n denotes the $n \times n$ identity matrix. Here we call $H(A) = (A + A^*)/2$ and $K(A) = (A - A^*)/2$ the hermitian and the skew-hermitian parts of $A = H(A) + K(A) \in \mathbb{C}^{n\times n}$, respectively. These two matrices have numerical ranges $F(H(A)) = \Re(F(A)) \subset \mathbb{R}$ and $F(K(A)) = \mathrm{i} \cdot \Im(F(A)) \subset \mathrm{i} \cdot \mathbb{R}$. For a modern view of $F(A)$ and its many properties, see [10, Chapter 1] for example.

*Received by the editors December 15, 2009. Accepted for publication March 7, 2010. Handling Editor: Michael J. Tsatsomeros.

†Department of Mathematics, National Technical University of Athens, Zografou Campus, 15780 Athens, Greece (horjoe@yahoo.gr, Ch. Chorianopoulos; ppsarr@math.ntua.gr, P. Psarrakos).

‡Department of Mathematics and Statistics, Auburn University, Auburn, AL 36849-5310, USA (uhligfd@auburn.edu).

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

Given a point $\mu \in F(A)$, we call a unit vector $x_\mu \in \mathbb{C}^n$ with $\mu = x_\mu^* A\, x_\mu$ a *generating vector* for μ. Motivated by the influence of the condition "$0 \in F(A)$" on the behavior of the systems $\dot{x} = A\,x$ and $x_{k+1} = A\,x_k$, Uhlig [13] posed the inverse numerical range (field of values) problem: *given an interior point μ of $F(A)$, determine a generating vector x_μ of μ.* He also proposed a complicated geometric algorithm that initially generates points of $F(A)$ that surround μ by using the fact that points on the boundary $\partial F(A)$ of $F(A)$ and their generating vectors can be computed by Johnson's eigenvalue method [11]. Then Uhlig's method [13] proceeds with a randomized approach in the attempt to surround the desired point μ tighter and tighter, and thereby iteratively refines the generating vector approximation.

In the sequel, Carden [2] observed the connection between the inverse numerical range problem and iterative eigensolvers, and presented a simpler iterative algorithm that relies on Ritz vectors as used in the Arnoldi method and yields an exact result for most points in a few iterations. In particular, his method is based on

(a) the iterative construction of three points of $F(A)$ that encircle μ and associated generating vectors, and

(b) the fact that given two points $\mu_1, \mu_2 \in F(A)$ and associated generating vectors, one can determine a generating vector for any convex combination of μ_1 and μ_2 by using a reduction to the 2×2 case.

The most expensive part of these iterative schemes to surround μ by numerical range points with known generators lies in the number of attempted eigen-evaluations, each at $O(n^3)$ cost for an $n \times n$ matrix A. Once μ is surrounded, the main remaining cost is associated with the number of $x^* A\, x$ or $x^* A\, y$ evaluations, each at only $O(n^2)$ cost.

In this note, we propose a much simpler geometric algorithm for solving the inverse numerical range problem than either of [13] or [2]. The new algorithm is faster and gives numerically accurate results where the previous methods often fail, such as when μ lies in very close distance from the actual numerical range boundary $\partial F(A)$, both if $\mu \in F(A)$ and $\mu \notin F(A)$. It differs from Carden's and Uhlig's original algorithms [2, 13] in the following ways:

(i) Rather than straight line interpolations of $F(A)$ boundary points that have been computed by Johnson's method [11], we use the ellipses that are the images $z^* A\, z$ when z traverses a great circle on the complex unit sphere of $\mathbb{C}^n$. These complex great circles on the unit sphere map to ellipses under $x \mapsto x^* A\, x$, as was established by Davis [3]. The use of ellipses instead of line segments that are derived from two computed $\partial F(A)$ points helps us obtain numerical range generators for points on both sides of μ more readily and results in a significant speed-up; see point (ii) below and Table 3 in the tests section.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

(ii) From two given points $\alpha_1, \alpha_2 \in F(A) \cap \Im(\mu) \cdot \mathbb{R}$ with $\Re(\alpha_1) \leq \Re(\mu) \leq \Re(\alpha_2)$ and their generating vectors, we compute a generating vector for μ by applying Proposition 1.1 below (see [10, page 25]) instead of using a reduction to the two-dimensional case as done in [2].

(iii) If in the initial two eigenanalyses of the hermitian matrices $H(A)$ and $-\mathrm{i}\,K(A)$, the subsequently generated ellipse intersections with the line $\Im(z) = \Im(\mu)$ do not give us a generating vector pair with numerical range points to the left and right of μ, then we bisect the relevant supporting angles in $A(\theta) = \cos(\theta)\,H(A) + \sin(\theta)\,\mathrm{i}\,K(A)$ and perform further eigenanalyses of $A(\theta)$ until the resulting great circle generated ellipses inside $F(A)$ either satisfy part (ii) above and we can solve the inverse problem by Proposition 1.1, or until one of the matrices $A(\theta)$ becomes definite. In the latter case, we conclude that $\mu \notin F(A)$ and we stop.

PROPOSITION 1.1. [10, page 25] *Let $A \in \mathbb{C}^{n\times n}$ be a matrix whose numerical range $F(A)$ is not a singleton, and let a and c be two real axis points of $F(A)$ with $a < 0 < c$. Suppose that $x_a, x_c \in \mathbb{C}^n$ are two unit vectors that generate $x_a^* A\, x_a = a$ and $x_c^* A\, x_c = c$.*

(a) *For $x(t,\vartheta) = e^{\mathrm{i}\vartheta} x_a + t\, x_c \in \mathbb{C}^n$, $\alpha(\vartheta) = e^{-\mathrm{i}\vartheta} x_a^* A\, x_c + e^{\mathrm{i}\vartheta} x_c^* A\, x_a$ and $t, \vartheta \in \mathbb{R}$, we have $x(t,\vartheta)^* A\, x(t,\vartheta) = c\,t^2 + \alpha(\theta)\,t + a$ and $\alpha(-\vartheta) \in \mathbb{R}$ when $\vartheta = \arg(x_c^* A\, x_a - x_a^T \overline{A}\, \overline{x_c})$.*

(b) *For $t_1 = \left(-\alpha(-\varphi) + \sqrt{\alpha(-\varphi)^2 - 4\,a\,c}\right)/(2\,c)$, we have*

$$x(t_1, -\varphi) \neq 0 \quad \text{and} \quad \frac{x(t_1,-\varphi)^*}{\|x(t_1,-\varphi)\|_2}\, A\, \frac{x(t_1,-\varphi)}{\|x(t_1,-\varphi)\|_2} = 0.$$

2. The algorithm. For $A \in \mathbb{C}^{n\times n}$ and μ an interior point of $F(A)$, we replace the problem of finding a unit vector $x \in \mathbb{C}^n$ with $x^* A\, x = \mu$ $(= x^* \mu\, I_n\, x)$ with the equivalent problem

$$x^*(A - \mu\, I_n)x = 0$$

immediately. Thus, without loss of generality, we assume that $\mu = 0$ and look for a unit vector x_0 such that $x_0^* A\, x_0 = 0$; i.e., we simply replace A by $A - \mu\, I_n$ if $\mu \neq 0$.

First we construct up to four $\partial F(A)$ points p_i and their generating unit vectors x_i $(i = 1, 2, 3, 4)$, by computing the extreme eigenvalues with associated unit eigenvectors x_i for $H(A) = (A + A^*)/2$ and $K(A) = (A - A^*)/2$. By setting $p_i = x_i^* A\, x_i$, we obtain four $F(A)$ points p_i that mark the extreme horizontal and vertical extensions of $F(A)$. These we denote by rM and rm for the maximal and minimal horizontal extension $F(A)$ points, respectively, and likewise by iM and im for the extreme vertical $F(A)$ points. If any one of these lies within 10^{-13} of zero in absolute terms,

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

then we accept the corresponding unit vector as the desired generating vector. If on the other hand, one of the hermitian matrices $H(A)$ and $\mathrm{i}\,K(A)$ is found to be definite during the eigenanalyses, then we stop knowing that $\mu \notin F(A)$.

Next we determine the real axis intersections of the great circle ellipses that pass through each feasible pair of our computed $\partial F(A)$ points $p_i = x^*Ax,\ p_j = y^*Ay$. Here, a feasible pair refers to their imaginary parts having opposite signs. If among these there are real axis points on both sides of zero, then we compute a generating unit vector for $0 \in F(A)$ by using Proposition 1.1, and the inverse problem is solved. Otherwise, we study the quadratic expression whose zeros determine the coordinate axes $F(A)$ points on the ellipses through the points $x^*Ax,\ y^*Ay \in \partial F(A)$ and that are generated by the points in $\mathbb{C}^n$ on the great circle through x and y. It is

$$(2.1)\quad (tx+(1-t)y)^*A(tx+(1-t)y) = (x^*Ax + y^*Ay - (x^*Ay + y^*Ax))\,t^2 + (-2\,y^*Ay + (x^*Ay + y^*Ax))\,t + y^*Ay.$$

This is a quadratic polynomial equation over the complex numbers, and we are interested only in the solutions whose imaginary parts are equal to zero in order to apply Proposition 1.1 if possible. Setting the imaginary part of expression (2.1) equal to zero leads to the following polynomial equation with real coefficients:

$$(2.2)\qquad t^2 + g\,t + \frac{p}{f} = 0$$

for $q = \Im(x^*Ax),\ p = \Im(y^*Ay)$ and $r = \Im(x^*Ay + y^*Ax)$, so that $f = p + q - r$ and $g = (r - 2\,p)/f$. Equation (2.2) has two real solutions $t_i,\ i = 1, 2$, and these supply two generating vectors $x_i = t_i\,x + (1 - t_i)\,y\ \ (i = 1, 2)$ for two real axis points. Normalization then gives two unit vector generators as desired. A set of great circle image points ($\bullet$) and their two real axis crossings ($+$) are depicted in Figure 1 for a specific matrix A that will be described in the next section.

If, unlike Figure 1, none of the feasible ellipses gives us two real axis numerical range points to either side of zero initially, then we check whether their collective set does. If not, we compute more eigenanalyses for $A(\theta) = \cos(\theta)\,H(A) + \sin(\theta)\,\mathrm{i}\,K(A)$ with angles θ other than $\theta = 0$ and $\theta = \pi/2$ as done at start-up with $H(A)$ and $\mathrm{i}\,K(A)$, respectively. If for example all original ellipses intersect the real axis to the right of zero and $\Im(rm) < 0$, then we bisect the third quadrant and compute the largest eigenvalue and associated eigenvector x_{new} of $A(3\pi/4)$ to find a $\partial F(A)$ point that lies between iM and rm. If this point lies below the real axis, then we check the ellipse intersections of the great circle images through the generating vector of iM and x_{new}, otherwise we do the same for the generator of rm and x_{new}. Thus, we proceed by angle bisection until we encounter a definite matrix $A(\theta)$ indicating that $\mu \notin F(A)$, or an ellipse that intersects the real axis to the left of zero and we can solve the inverse problem by using Proposition 1.1. The number of iterations by

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

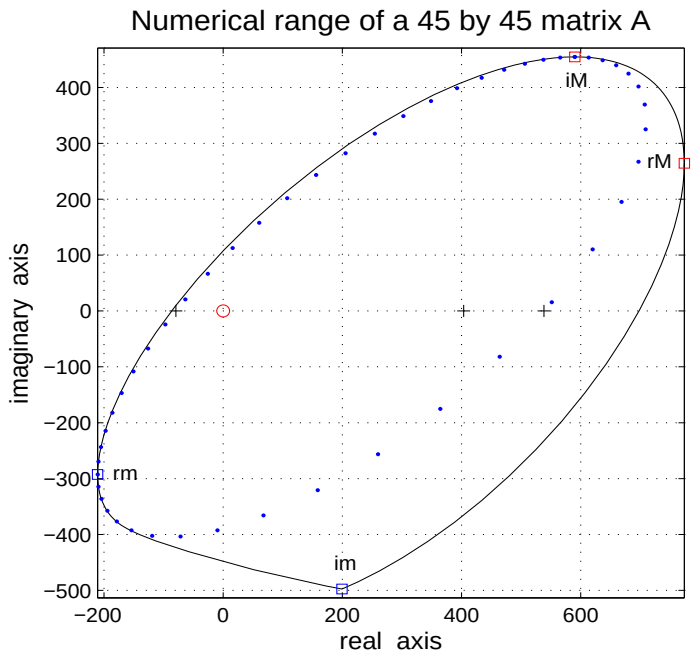

FIGURE 2.1. *Numerical range boundary* (–) *with extreme horizontal extension* $\partial F(A)$ *points* rm (□) *and* rM (□)*, and extreme vertical extensions of* $F(A)$ *at* iM (□) *and* im (□)*. Image points* (•) *of the great circle through the generating vectors of* iM *and* rm*, and its real axis crossings* (+) *(and an earlier one). Zero in* $\mathbb{C}$ *at* o*.*

angle bisection has generally been low, normally below 4 and possibly up into the teens only when zero lies within 10^{-13} of the boundary $\partial F(A)$ in absolute terms.

Overall, the most expensive parts of our algorithm are the eigenanalyses of $A(\theta)$ at $O(n^3)$ cost. The remainder of the program requires a number of quadratic form evaluations in the form of $x^* A\, y$ as in Proposition 1.1(a) and in equations (2.1) and (2.2) at $O(n^2)$ cost, as well as sorts, finds etc at even lower $O(n)$ or $O(1)$ computational cost.

3. Tests. The 45×45 complex matrix whose numerical range is depicted in Figure 2.1 is constructed from the complex matrix $B \in \mathbb{C}^{45\times 45}$ in [9, p. 463]. This matrix B is the sum of the Fiedler matrix $F = (|i-j|)$ and i times the Moler matrix $M = U^T U$ for the upper triangular matrix U with $u_{i,j} = -1$ for $j > i$; i.e., $B = F + \mathrm{i}\, M$. We obtain A for Figure 2.1 by adding $-3 + \mathrm{i}\, 5$ times the matrix of all ones and shifting by $\mu = -200 + \mathrm{i}\, 500$, or in MATLAB notation, `A=B+(-3+5i)*ones(45)-(-200+500i)*eye(45)`. Here are some run comparisons for

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

the algorithms of [2, 13] and our algorithm:

$n = 45$	execution time	eigenanalyses	error $\lvert x^* A\, x - 0\rvert$
`wberpoint` from [13]	0.1 to 0.15 sec	3	10^{-10} to $9 \cdot 10^{-11}$
`inversefov` from [2]	0.0071 sec	3	$3.6 \cdot 10^{-13}$
this paper's algorithm	0.0042 sec	2	$2.3 \cdot 10^{-13}$

Table 1

Note that `wberpoint` relies on randomly generated vectors in $\mathbb{C}^n$ and therefore its run data will vary with the chosen random vectors accordingly. For the same type of 500×500 matrix B, we consider the matrix generated by `A=B+(-3+5i)*ones(500)-(-200+500i)*eye(500)` and denote it by $A_{500,500}$. The computed data is as follows:

$n = 500$	execution time	eigenanalyses	error $\lvert x^* A\, x - 0\rvert$
`wberpoint` from [13]	2.3 to 4.8 sec	0 to 1 (`eigs`)	$5 \cdot 10^{-10}$ to $4 \cdot 10^{-12}$
`inversefov` from [2]	0.75 sec	2 (`eig`)	$3.7 \cdot 10^{-11}$
this paper's algorithm	0.24 sec	4 (`eigs`)	$6 \cdot 10^{-13}$

Table 2

The 500×500 matrix $A_{500,500}$ has one tight cluster of 493 relatively small eigenvalues of magnitudes around 540 and 7 separate eigenvalues of increasing magnitudes up to $11.2 \cdot 10^4$. Our inverse numerical range routine as well as those of [2, 13] use the Krylov type eigensolver `eigs` in MATLAB for large dimensional matrices A for its better speed in finding the largest and/or smallest eigenvalue(s) of $A(\theta)$ that we need. Since all iteration matrices $A(\theta) = \cos(\theta)\, H(A) + \sin(\theta)\, \mathrm{i}\, K(A)$ are hermitian, their eigenvalues are well conditioned. But their tight clustering and large magnitude discrepancies make the Lanczos method of `eigs` quite unsuitable when used with the MATLAB default setting for the Ritz estimate residual tolerance `opts.tol` to the machine constant `eps` $\doteq 2.2 \cdot 10^{-16}$. To gain convergence of `eigs` in our algorithm for $A_{500,500}$ we have to use `opts.tol` values between 10^{-5} and 10^{-3} instead. This apparently has no ill effect on our inverse problem accuracy. Likewise changing the `opts.tol` setting inside the code of [2] does, however, not help since its call of `eigs` crashes. Therefore, the current code `inversefov` in [2] is of limited use in clustered eigen-situations. On the other hand, [2] works well as does ours, when using the much more expensive Francis QR based dense eigensolver `eig` of MATLAB. Note that our code with its very loose Lanzcos based eigensolver still obtains a generating vector for μ that is almost two orders of magnitude better than what `eig` can achieve in `inversefov` from [2]. Finally, if we use our code with `eig` instead for this $A_{500,500}$, the run time will go up comparably but the error comes in at only around $3.5 \cdot 10^{-12}$.

Our next example features a 10×10 matrix A generated by the MATLAB command `A= randn(10)+(3+3i)*ones(10)`. A's numerical range is elongated as shown

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

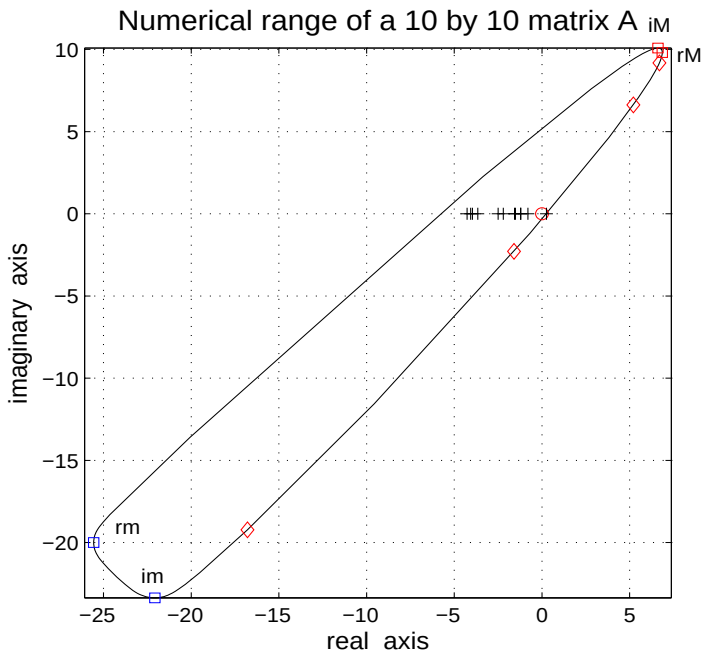

FIGURE 3.1. *Numerical range boundary* (–) *with extreme horizontal extension* $\partial F(A-\mu I_{10})$ *points rm* (□) *and rM* (□)*, and extreme vertical extensions at iM* (□) *and im* (□)*. Numerical range real axis points* (+)*: several to the left and one to the right of zero, as well as intermediate* $\partial F(A)$ *points* (◊) *of* 4 *angle bisecting steps.*

in Figure 3.1, where we have shifted the origin by $\mu = 22.5 + \mathrm{i}\,20$.

For the chosen $\mu = 22.5 + \mathrm{i}\,20$, our method finds a generating vector after 4 angle bisections since according to Figure 3.1 then there are known generating vectors for real axis points to the left and right of zero and Proposition 1.1 applies. If we move μ closer to the edge of $F(A)$ by increasing $\Re(\mu)$, the computed data for this example is as follows:

μ	C number	`inversefov` from [2]			this paper's algorithm		
		sec	eig	error	sec	eig	error
$22.83543 + \mathrm{i}\,20$	$-5 \cdot 10^{-8}$	0.09	12	$8 \cdot 10^{-13}$	0.045	14	10^{-15}
$22.835430065 + \mathrm{i}\,20$	$-3 \cdot 10^{-10}$	0.11	17	$5 \cdot 10^{-12}$	0.045	14	$3.6 \cdot 10^{-15}$
$22.8354300651 + \mathrm{i}\,20$	$-2.4 \cdot 10^{-10}$	0.21	35	$3 \cdot 10^{-12}$	0.047	14	10^{-15}
$22.835430065417 + \mathrm{i}\,20$	$-7 \cdot 10^{-13}$	*	*	*	0.049	16	10^{-15}
$22.835430065418 + \mathrm{i}\,20$	$4 \cdot 10^{-13}$	*	*	*	μ outside $F(A)$		

Table 3

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

Here the 'C number' denotes the generalized Crawford number and indicates the minimal distance of μ from the boundary of the numerical range of $A - \mu I_{10}$. It was computed via `Craw2circ.m` of [15, 16]. As long as the generalized Crawford number of a given point $\mu \in \mathbb{C}$ is negative, the point lies inside the numerical range. The third data row of the Table 3 exhibits the best that could be achieved with `inversefov` from [2]; if μ is changed to $22.8354300652 + \mathrm{i}\,20$ for example, `inversefov` fails altogether while our algorithm works correctly for three more digits or 15 digit precision in μ. Note how the origin approaches $\partial F(A)$ in successive data rows from order 10^{-8} to order 10^{-13}, and that in our algorithm the number of eigenanalyses hardly increases here. The 5th data line above indicates that our algorithm determined that the point $\mu = 22.835430065418 + \mathrm{i}20$ to lie outside the numerical range of A with a distance of order $4 \cdot 10^{-13}$ from its boundary. This also took 16 eigenanalyses.

Our last example comes from [13, Example 7] and uses a 188×188 Jordan block J for the eigenvalue $1 + \mathrm{i}\,3$ (with all co-diagonals equal to 1) as test matrix. Here, both [2] and our algorithm use 3 eigenanalyses; our algorithm finds a generating unit vector for the point $\mu = 1.707 + \mathrm{i}\,3.707$ that lies inside $F(A)$ within 10^{-5} of $\partial F(A)$ in 0.17 seconds with accuracy of order 10^{-17} while `inversefov` from [2] uses 0.2 seconds with $9 \cdot 10^{-16}$ accuracy. For this example, `wberpoint` of [13, Example 7] achieves an accuracy of $3.8 \cdot 10^{-10}$ in almost 1 second and uses 7 eigenanalyses.

4. Conclusions and comments. We have developed, implemented and tested a short, quick and accurate inverse numerical range algorithm that finds a generating complex unit vector of any field of values point in its interior. The proposed method relies on geometric ideas for computations in the realm of the matrix numerical range and is frugal with its $O(n^3)$ eigenanalyses that are – at this time – apparently necessary to proceed here.

To also look for imaginary axis field of values points below and above μ in the complex plane and their generators might offer another, quite natural improvement as well, but first tries with a thus expanded version of our subroutine `Cvectinterpol` inside [14] have shown but little speed-up, if any, and that idea has been abandoned.

Finally, we remark that our code was tested using MATLAB2009b on both MACs and PCs.

REFERENCES

[1] C.A. Beattie, M. Embree, and D.C. Sorensen. Convergence of polynomial restart Krylov methods for eigenvalue computations. *SIAM Rev.*, 47:492–515, 2003.

[2] R. Carden. A simple algorithm for the inverse field of values problem. *Inverse Problems*, 25:115019, 2009.

Electronic Journal of Linear Algebra ISSN 1081-3810
A publication of the International Linear Algebra Society
Volume 20, pp. 198-206, March 2010

ELA
http://math.technion.ac.il/iic/ela

[3] C. Davis. The Toeplitz-Hausdorff Theorem explained. *Canad. Math. Bull.*, 14:245–246, 1971.
[4] M. Eiermann. Field of values and iterative methods. *Linear Algebra Appl.*, 180:167–197, 1993.
[5] M. Embree and L.N. Trefethen. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*. Princeton University Press, 2005.
[6] M. Goldberg and E. Tadmor. On the numerical radius and its applications. *Linear Algebra Appl.*, 42:263–284, 1982.
[7] K.E. Gustafson and D.K.M. Rao. *Numerical Range, The Field of Values of Linear Operators and Matrices*. Springer-Verlag, New York, 1997.
[8] F. Hausdorff. Der Wertvorrat einer Bilinearform. *Math. Z.*, 3:314–316, 1919.
[9] N.J. Higham, F. Tisseur, and P.M. van Dooren. Detecting a definite hermitian pair and a hyperbolic or elliptic quadratic eigenvalue problem, and associated nearness problems. *Linear Algebra Appl.*, 351/352:455–474, 2002.
[10] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1991.
[11] C.R. Johnson. Numerical determination of the field of values of a general complex matrix. *Linear Algebra Appl.*, 15:595–602, 1978.
[12] O. Toeplitz. Das algebraische Analogon zu einern Satze von Fejér. *Math. Z.*, 2:187–197, 1918.
[13] F. Uhlig. An inverse field of values problem. *Inverse Problems*, 25:055019, 2008.
[14] F. Uhlig. MATLAB m-file `invfovCPU.m`, 2010. Available at http://www.auburn.edu/~uhligfd/m_files/invfovCPU.m.
[15] F. Uhlig. On computing the generalized Crawford number of a matrix. *Submitted*.
[16] F. Uhlig. MATLAB m-file `Craw2circ.m`, 2010. Available at http://www.auburn.edu/~uhligfd/m_files/Craw2circ.m.