MARYLAND

TECHNICAL
RESEARCH
REPORT

SRC TR 88 - 36

STRUCTURED SINGULAR VALUE and m-FORM NUMERICAL RANGE

by

Jia-Chang Wang

SYSTEMS RESEARCH CENTER

UNIVERSITY OF MARYLAND

COLLEGE PARK, MARYLAND 20742

SRC Library
PLEASE DO NOT REMOVE
Thank You

STRUCTURED SINGULAR VALUE AND m-FORM NUMERICAL RANGE

by

Jia-Chang Wang

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1988

APPROVAL SHEET

Title of Thesis: Structured Singular Value and
m-Form Numerical Range

Name of Candidate: Jia-Chang Wang
Master of Science, 1988

Thesis and Abstract Approved:

Dr. Michael K.H. Fan
Assistant Research Scientist,
Systems Research Center,
University of Maryland, College Park

Date Approved: 3/3/88

ABSTRACT

Title of Thesis: Structured Singular Value and the m-Form Numerical Range

Jia-Chang Wang, Master of Science, 1988

Thesis directed by: Dr. Michael K.H. Fan
Assistant Research Scientist
Systems Research Center,
University of Maryland, College Park

Although the question of the numerical evaluation of Doyle's structured singular value has been repeatedly addressed, it is not yet entirely resolved in the general case. It has been shown recently that this question can be reduced to that of iteratively computing the distance from the origin to the m-form numerical range of m tuples of matrices. How to effectively compute such distance in the nonconvex case (which may arise when dealing with m more than 3) is an open problem. In this thesis, in an attempt to tackle this problem, the question of graphically displaying 2-dimensional and 3-dimensional sections of the m-form numerical range is investigated.

STRUCTURED SINGULAR VALUE AND m-FORM NUMERICAL RANGE

by

Jia-Chang Wang

Thesis submitted to the Faculty of the Graduate School
of The University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1988

Advisory Committee:

Assistant Research Scientist Dr. Michael K.H. Fan
Associate Professor Dr. André L. Tits
Professor Dr. P.S. Krishinaprasad
Associate Professor Dr. Mark Shayman

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to my thesis advisor, Dr. Machael K.H. Fan, for his continuing guidance and encouragement during the course of this research.

Also, I would like to thank Dr. André L. Tits for his sopport, stimulating discussing and patient in correcting my writing, and Dr. P.S. Krishnaprasad and Dr. Mark Shayman for their serving im my thesis examining committee.

This research is supported by the National Science Foundation under Grants No. DMC-84-51515 and CDR-85-00108.

Finally, I am grateful to thank my family for their mornal support and encouragement.

TABLE OF CONTENTS

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 Outline of Thesis 4

1.3 Structured Singular Value 4

1.4 Numerical Range and m-Form Numerical Range 7

1.5 Some Properties of the m-Form Numerical Range 8

1.6 Structured Singular Value and m-Form Numerical Range 10

Chapter 2 Graphically Displaying m-Form Numerical Range . 15

2.1 Introduction 15

2.2 Two-Dimensional Case 16

2.3 Coordinate Transformations 19

2.4 Two-Dimensional Sections of W 21

2.5 Non-Convex Boundary of W 25

2.6 Three-Dimensional Sections of W 33

Chapter 3 Conclusion 37

Appendex A Software Package 39

A.1 Introduction 39

A.2 Program mait 40

A.3 Program nr 44

A.4 Programs bf and fbd 47

A.5 Program grap 48

A.6 Program tisosrf 51

Appendex B Doyle's Example 52

References 54

LIST OF FIGURES

1. General Framework 2
2. Example of Procedure 2.1 18
3. Generated Boundary Points of Figure 2 19
4. Property of Curvature 20
5. Convex Hull of Doyle's Example for $W(\mu_K^2(M))$ 23
6. A Family of 2-Dimensional Curves 23
7. Doyle's Example for $W(\mu_K^2(M))$ by Brute Force 25
8. Boundary Points of Figure 7 26
9. Non-Convex Section of Doyle's Example 28
10. 28
11. 29
12. 29
13. 30
14. 30
15. 31
16. 31
17. 3-Dimensional Convex Surface 35
18. 3-Dimensional Non-Convex Surface of Doyle's Example 36
19. Different View Poisition of Figure 18 36

CHAPTER

ONE

Introduction

1.1 Motivation

The concept of structured singular value was introduced by Doyle [1]. It was motivated by simultaneously analyzing the robust performance and robust stability properties of feedback systems affected by possibly structured uncertainty. The general framework to be used in this analysis is illustrated in the diagram in Figure 1. $P(s)$ represents all the system dynamics including interconnection of nominal plants, controllers, sensors, and any scalings etc., and is assumed to be described by a causal, stable and real proper transfer function, $e(s)$ is the error of the system, $\Delta(s)$ is the perturbations of the system, and $u(s)$ is the input of the system respectively. Doyle [2,3] showed that any linear interconnection of inputs, outputs, commands, perturbations and controllers can be rearranged to match this diagram. The analysis problem involves determining whether the error e remains in a desired set for sets of input u and perturbation Δ. It is convenient to partition $P(s)$ into the form

$$P(s) = \begin{bmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{bmatrix} ,$$

where $P_{ij}(s)$ corresponding to (vector) input j and (vector) output i in Figure 1. The (casual) plant uncertainty $\Delta(s)$ has the form

$$\Delta(s) = \text{block diag } \{\delta_1 I_{k_1}, \cdots, \delta_r I_{k_r}, \Delta_1(s), \ldots, \Delta_c(s)\} ,$$

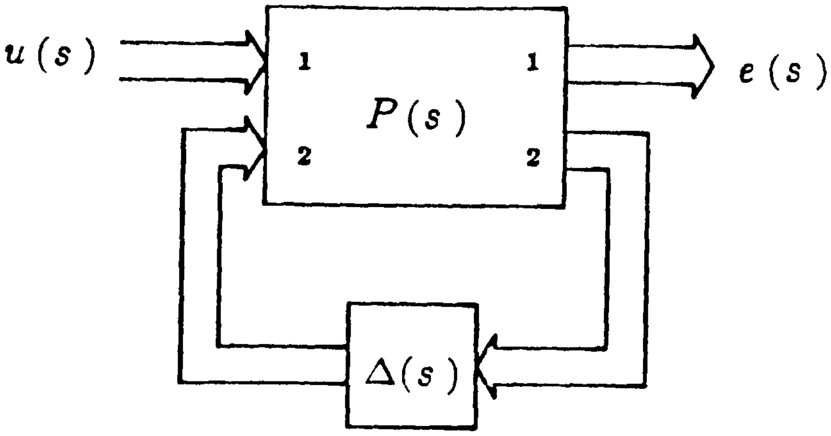

Figure 1: General Framework

where $r + c$ specifies the number of blocks of the simultaneous uncertainties. Thus, uncertainty is assumed to be localized in some specific subsystems. In practical problem, it is generally the case that the uncertainty consists of parameter variations and multiple norm-bounded perturbations which correspond to the constant real scalars δ_i and to the submatrices $\Delta_i(s)$ respectively. Parameter variations typically arise because of uncertain coefficients in differential equation models of physical systems and involve real scalars. Norm-bounded perturbations often arise when trying to capture the effect of unmodeled dynamics and are themselves dynamical systems. The $\Delta_i(s)$'s are assumed to have known bounds over the closed right half complex plane C_+ but to be otherwise unknown, and the δ_i's are assumed to be real numbers within known bounds but otherwise unknown. The external input $u(s)$ is an additive signal entering the system and is typically used to model disturbances, commands and noises. It is conventional to absorb any scalings, weights, or coloring filter into P so that u, e, and $\Delta(s)$ are normalized to

$$\sup_{\omega \geq 0} \bar{\sigma}(\Delta(j\omega)) < 1 \tag{1.1}$$

where $\bar{\sigma}(\cdot)$ is the largest singular value of its matrix-valued argument, and

$$\int_0^\infty u^T(t)u(t)dt < 1 \tag{1.2}$$

where, with a slight abuse of notation, $u(t)$ (resp. $e(t)$) represents the inverse Laplace transformation of $u(s)$ (resp. $e(s)$). The input-output mapping in the diagram in Figure 1 is

$$e(s) = [P_{11}(s) + P_{12}(s)\Delta(s)(I - P_{22}(s)\Delta(s))^{-1}P_{21}(s)]u(s) \ .$$

The introduction by Doyle [1] of the structured singular value was motivated by an attempt to determine when the following analysis criterion holds.

Analysis Criterion 1. The feedback system in Figure 1 is stable and the error e satisfies

$$\int_0^\infty e^T(t)e(t)dt < 1$$

for all input $u(s)$ and bounded uncertainty $\Delta(s)$ for which (1.1) and (1.2) hold. □

The question then is whether a necessary and sufficient condition can be imposed on the transfer function $P(s)$ to ensure fulfillment of Analysis Criterion 1. To address this question, Doyle introduced a new measure for matrices, the structured singular value denoted by μ. This new measure characterizes the largest block-structured uncertainty which makes Analysis Criterion 1 satisfied.

1.2 Outline of Thesis

Although the question of the evaluation of Doyle's structured singular value has been repeatedly addressed, it is not yet entirely resolved in the general case (see section 1.6). It has been shown recently that this question can be reduced to that of iteratively computing the distance from the origin to the m-form numerical range of certain tuples of matrices, where m is the number of 'blocks' of the structure [4]. How to effectively compute such distance in the non-convex case (which may arise when m is greater than 3) is an open problem. In this thesis, in an attempt to tackle this problem, we examine the properties of the m-form numerical range and propose procedures to graphically display the boundaries of its 2-dimensional and 3-dimensional sections. In Chapter 1, we define the structured singular value, the numerical range, the m-form numerical range and then discuss some of their properties. In Chapter 2, we investigate the question of graphically displaying sections of the m-form numerical range. Specially, we first propose several procedures to display any specified 2-dimensional section. By employing the NCAR [5] graphics software and the data generated by the above procedures, we then propose a procedure to display the surface of the corresponding 3-dimensional section. For the non-convex case, we show several interesting pictures of Doyle's example (see Appendix B). We hope this geometric insight will be helpful for the question of computing the structured singular value.

1.3 Structured Singular Value

Throughout the thesis, we use following notations:

1. Given any square complex matrix M, we denote by $\rho(M)$ its spectral radius, by $\overline{\sigma}(M)$ its largest singular value, by M^H its complex conjugate

transpose, and define $\rho_R(M)$ as $\rho_R(M) = 0$ if M has no real eigenvalue, $\rho_R(M) = \max\{|\lambda| \ : \ \lambda \text{ is a real eigenvalue of } M\}$ otherwise.

2. Given any complex vector x, x^H denotes its complex conjugate transpose and $\|x\|$ its Euclidian norm.

3. $\partial \mathrm{B}$ denotes the unit sphere in $\mathbb{C}^n$, i.e., $\partial \mathrm{B} = \{x \in \mathbb{C}^n, \|x\| = 1\}$.

4. For any positive integer k, we denote by O_k the $k \times k$ zero matrix, and by I_k the $k \times k$ identity matrix.

5. Given an $n \times n$ complex matrix M and 2 nonnegative integers r and c, a block-structure $\mathcal{K}$ of dimensions (r, c) associated with M is an $(r+c)$-tuple of positive integers

   $$\mathcal{K} = (k_1, \ldots, k_r \ ; \ k_{r+1}, \ldots, k_{r+c})$$

   such that $\sum_{q=1}^{r+c} k_q = n$.

6. Given a block-structure $\mathcal{K}$ of size $r + c$, we make use of the family of block diagonal matrices with any positive scalar δ (possibly ∞)

   $$\begin{aligned} X_{\mathcal{K}}(\delta) = \ & \{\text{block diag}(\delta_1 I_{k_1}, \ldots, \delta_r I_{k_r}, \Delta_1, \ldots, \Delta_c) \ : \\ & \delta_i \in \mathrm{I\!R}, \ |\delta_i| \leq \delta, \Delta_i \in \mathbb{C}^{k_{r+i} \times k_{r+i}}, \bar{\sigma}(\Delta_i) \leq \delta\}; \end{aligned}$$

   of the family of block diagonal matrices

   $$\begin{aligned} \mathcal{D}_{\mathcal{K}} = \ & \{\text{block diag}(D_1, \ldots, D_r, d_1 I_{k_{r+1}}, \ldots, d_c I_{k_{r+c}}) \ : \\ & D_i \in \mathbb{C}^{k_i \times k_i}, D_i = D_i^H > 0, \ d_i \in (0, \infty)\}, \end{aligned}$$

   (if A is Hermitian, $A > 0$ expresses that A is positive definite); of the family of block diagonal unitary matrices

   $$\begin{aligned} \mathcal{U}_{\mathcal{K}} = \ & \{\text{block diag}(u_1 I_{k_1}, \ldots, u_r I_{k_r}, U_1, \ldots, U_c) \ : \\ & u_i \in \{-1, 1\}, U_i \in \mathbb{C}^{k_{r+i} \times k_{r+i}}, U_i U_i^H = I\} \ . \end{aligned}$$

Definition 1.1. [1] The *structured singular value* $\mu_{\mathcal{K}}(M)$ of a $n \times n$ complex matrix M with respect to block-structure $\mathcal{K}$ is the positive number μ having the property that

$$\det(I + M\Delta) \neq 0 \text{ for all } \Delta \in X_{\mathcal{K}}(\delta)$$

if, and only if

$$\delta\mu < 1 .$$

in other words, $\mu_{\mathcal{K}}(M) = 0$, if there is no Δ in $X_{\mathcal{K}}(\infty)$ such that $\det(I+M\Delta) = 0$, and $(\min_{\Delta \in X_{\mathcal{K}}(\delta)}\{\overline{\sigma}(\Delta) \ : \ \det(I + M\Delta) = 0\})^{-1}$ otherwise. □

The structured singular value yields a necessary and sufficient condition for Analysis Criterion 1 to be satisfied. We state the result in the following fact [2].

Fact 1.1. [2] Analysis Criterion 1 holds if, and only if,

$$\sup_{\omega \geq 0} \mu_{\mathcal{K}}(P(j\omega)) \leq 1 ,$$

where $j = \sqrt{-1}$, and $\mathcal{K}$ is the block-structure of $\Delta(s)$ augmented by adding one complex block with the size equal to the number of u (or e). □

The structured singular value satisfies the following properties [6].

$$\max_{U \in \mathcal{U}_{\mathcal{K}}} \rho_R(MU) \leq \mu_{\mathcal{K}}(M) . \tag{1.3}$$

The inequality in (1.3) becomes an equality, with ρ replacing ρ_R, if $r = 0$ (no real uncertainty).

$$\mu_{\mathcal{K}}(M) \leq \inf_{D \in \mathcal{D}_{\mathcal{K}}} \bar{\sigma}(DMD^{-1}) . \tag{1.4}$$

For $r = 0, c \leq 3$ or $r = 1, c = 1$

$$\mu_{\mathcal{K}}(M) = \inf_{D \in \mathcal{D}_{\mathcal{K}}} \bar{\sigma}(DMD^{-1}) \ . \tag{1.5}$$

Doyle [7] exhibited an example with block-structure $\mathcal{K} = (;1,1,1,1)$, for which equality in (1.5) does not hold. Using (1.3) to compute the structured singular value is discarded in [1] because the optimization problem it involved generally has nonglobal maxima. It should be noted that any nonglobal maxima of $\max_{U \in \mathcal{U}} \rho_R(MU)$ is a lower bound, i.e., an optimistic bound, for $\mu_{\mathcal{K}}(M)$. On the other hand, (1.4) always generate an upper bound of $\mu_{\mathcal{K}}(M)$, which is conservative and does guarantee the robust stability (here robust is used to indicate that the property of stability is maintained under perturbations). Algorithms [1,8] are available for this problem yielding a reliable way of computing $\mu_{\mathcal{K}}(M)$ for the cases when (1.5) holds. However, to date there is no reasonably efficient algorithm for the computation of $\mu_{\mathcal{K}}(M)$ in the general case.

1.4 Numerical Range and m-Form Numerical Range

The idea of the numerical range of an $n \times n$ complex matrix A was introduced by Toeplitz in 1918 [9]. It is the set of complex numbers defined by

$$W(A) = \{x^H A x \ : \ x \in \mathbb{C}^n, \ \|x\| = 1\} \ .$$

Toeplitz [9] showed that $\partial W(A)$ (the boundary of $W(A)$) is a convex curve, and a short time later Hausdorff [10] proved that $W(A)$ itself is convex. This result is known as Toeplitz-Hausdorff Theorem. The followings are some properties of the numerical range.

1. $W(A)$ is convex [9,10].

2. The non-differentiable points on the boundary of $W(A)$ are eigenvalues of A [11].

Any $n \times n$ complex matrix A can be decomposed into

$$A = \frac{A + A^H}{2} + j\frac{A - A^H}{2j} = A_1 + jA_2 \ ,$$

with A_1 and A_2 Hermitian and $j = \sqrt{-1}$ (in this thesis, we denote $\sqrt{-1}$ by j, while j is used as a running index). Thus we can view the numerical range $W(A)$ as a subset of $\mathbb{R}^2$:

$$W(A_1, A_2) = \left\{ \begin{bmatrix} x^H A_1 x \\ x^H A_2 x \end{bmatrix} \ : \ x \in \partial\mathrm{B} \right\} \ .$$

This representation suggests a generalization to more than two Hermitian forms (see Hausdorff [10], Brickman [12], and Au-Yeung [13]). Following [14], we now call this generalization to m matrices the m-form numerical range.

Definition 1.2. The m-form numerical range of an m-tuple of $n \times n$ Hermitian matrices $A_1, \ldots, A_m$ is the set

$$W(A_1, \ldots, A_m) = \{f(x) \ : \ \|x\| = 1, \ x \in \mathbb{C}^n\}$$

where $f : \mathbb{C}^n \longrightarrow \mathbb{R}^m$ has components

$$f^i(x) = x^H A_i x, \text{ for } i = 1, \ldots, m.$$

□

Clearly the numerical range is a special case of the m-form numerical range.

1.5 Some Properties of the m-Form Numerical Range

Given m complex Hermitian $n \times n$ matrices $A_1, \ldots, A_m$, their m-form numerical range $W = W(A_1, \ldots, A_m)$ satisfies the following properties.

Property 1.1. [13–16] If $n = 2$ and $m = 3$, W is (the boundary of) an ellipsoid. In all other cases, if $m \leq 3$, W is a convex. If the A_i's are real symmetric then, if $m \leq 3$, W is always convex. □

Property 1.2. [14] The intersection of $W(A_1, \ldots, A_m)$ with any of its supporting hyperplanes is an $\mathbb{R}^m$-embedding of the m-form numerical range of a certain $(m-1)$-tuple of matrices. □

The next property is an immediate consequence of Properties 1.1 and 1.2.

Property 1.3. If $m = 4$ and $A_1, \ldots, A_4$ are real symmetric, then W has a convex boundary, i.e., the intersection of W with any of its supporting hyperplanes is convex. □

Property 1.4. [14] If $x \in \partial\mathrm{B}$ is such that $f(x)$ is on $\partial W(A_1, \ldots, A_m)$ then there exists a direction $w \in \mathbb{R}^m$ such that x is an eigenvector of $\sum_{k=1}^{m} w^k A_k$. Moreover (i) if H is any supporting hyperplane to $W(A_1, \ldots, A_m)$ at $f(x)$, then the direction orthogonal to H is a valid choice for w. (ii) if for some $q \in \{1, \cdots, m\}$ there exists no subset of $W(A_1, \ldots, A_m)$ containing $f(x)$ that is locally homeomorphic to $\mathbb{R}^{m-q}$ around $f(x)$, then there is a $(q+1)$-dimensional subspace $\mathcal{S}$ of $\mathcal{V} = \{A \in \mathbb{C}^{n \times n} : A = \sum_{k=1}^{m} w^k A_k, w^k \in \mathbb{R}\}$ such that all matrices in $\mathcal{S}$ admit x as an eigenvector. □

Property 1.4 plays an important role in graphically displaying the boundary of the m-form numerical range. In the convex case, the normal vector of any supporting hyperplane is a valid choice of w such that $f(x)$ is a boundary point of W, where x is the unit eigenvector corresponding to the smallest eigenvalue of $\sum_{k=1}^{m} w^k A_k$ (see Section 2.2). For non-convex case, boundary point $f(x)$ need not be contact point for any supporting hyperplane of W, but the corresponding x still a unit eigenvector of $\sum_{k=1}^{m} w^k A_k$ for some w.

1.6 Structured Singular Value and m-Form Numerical Range

In Section 1.3, we gave the definition of the structured singular value $\mu_{\mathcal{K}}(M)$ of an $n \times n$ complex matrix M associated with a block-structure $\mathcal{K}$. In order to connect the structured singular value problem to the m-form numerical range, we review an equivalent expression of the structured singular value [6]. Given an $n \times n$ complex matrix M and associated block-structure $\mathcal{K} = (k_1, \ldots, k_r; k_{r+1}, \ldots, k_{r+c})$, we will make use of the projection matrices P_q, $q = 1, \ldots, r+c$,

$$P_q = \text{block diag}(O_{k_1}, \ldots, O_{k_{q-1}}, I_{k_q}, O_{k_{q+1}}, \ldots, O_{k_{r+c}}) \ ;$$

and for $q = 1, \ldots, r+c$, of the 'set of indices in block q'

$$J_q = \left\{ \sum_{j=1}^{q-1} k_j + 1, \sum_{j=1}^{q-1} k_j + 2, \ldots, \sum_{j=1}^{q} k_j \right\} \ ,$$

with the convention that $\sum_1^0 = 0$. For $i, j \in \{1, \ldots, \sum_{q=1}^{r} k_q\}$, define the $n \times n$ matrices E^{ij} by

$$(E^{ij})_{kl} = \begin{cases} 1 & \text{if } k = i \text{ and } l = j \\ 0 & \text{otherwise} \end{cases} \ ;$$

and define the set $\mathcal{S}_{\mathcal{K}}(M)$ by

$$\mathcal{S}_{\mathcal{K}}(M) = \left\{ x \in \partial\mathrm{B} \ : \ x^H E^{ij} M x = x^H M^H E^{ij} x, \forall (i,j) \in \bigcup_{q=1}^{r} J_q \times J_q \right\} .$$

Fact 1.2. [6] For any $n \times n$ complex matrix M and associated block-structure $\mathcal{K}$,

$$\mu_{\mathcal{K}}(M) = \begin{cases} 0 & \text{if } \mathcal{S}_{\mathcal{K}}(M) = \emptyset \\ \max\limits_{\substack{\theta \in \mathbb{R} \\ x \in \mathcal{S}_{\mathcal{K}}(M)}} \{\theta \ : \ \|P_q M x\| \geq \theta \|P_q x\|, \ q = 1, \ldots, r + c\} & \text{otherwise} \end{cases}$$

□

Fact 1.2 gives an alternative formula to compute the structured singular value. Now consider the family of elementary Hermitian matrices [6]

$$\begin{aligned} \mathcal{E} = \ & \{(E^{ij} + E^{ji}) \ : \ (i,j) \in \textstyle\bigcup_{q=1}^{r} J_q \times J_q, \ i \leq j\} \cup \\ & \{j(E^{ij} - E^{ji}) \ : \ (i,j) \in \textstyle\bigcup_{q=1}^{r} J_q \times J_q, \ i < j\} \ . \end{aligned}$$

Then $\mathcal{S}_{\mathcal{K}}(M)$ can be written as

$$\mathcal{S}_{\mathcal{K}}(M) = \{x \in \partial \mathrm{B} \ : \ x^H E M x = x^H M^H E x, \ \forall E \in \mathcal{E}\} \ .$$

Let us pick up an order for the entries in $\mathcal{E}$, so that

$$\mathcal{E} = \{E_1, \ldots, E_s\}$$

with $s = \sum_{q=1}^{r} k_q^2$. For $q = 1, \ldots, r + c$ and $\alpha \in \mathbb{R}$, let

$$A_q(\alpha) = \alpha P_q - M^H P_q M \tag{1.6}$$

and for $q = r + c + 1, \ldots, r + c + s$, let

$$A_q(\alpha) = j(E_q M - M^H E_q) \tag{1.7}$$

where in (1.7) the argument α is used for sake of uniformity. Then $A_q(\alpha) = A_q^H(\alpha)$, for $q = 1, \ldots, r + c + s$, and

$$\mu_{\mathcal{K}}(M) = \begin{cases} 0 & \text{if } \mathcal{S}_{\mathcal{K}}(M) = \emptyset \\ \max\limits_{\substack{\alpha \in \mathbb{R} \\ x \in \partial \mathrm{B}}} \{\alpha \ : \ x^H A_q(\alpha) x \leq 0, \ q = 1, \ldots, r + c; \\ \quad x^H A_q(\alpha) x = 0, \ q = r + c + 1, \ldots, r + c + s\} & \text{otherwise} \end{cases}$$

Consider the $(r+c+s)$-form numerical range associated with matrices $A_i(\alpha)$'s, for $i = 1, \ldots, r+c+s$, i.e.,

$$W(\alpha) = \{v \in \mathbb{R}^{r+c+s} \ : \ \exists x \in \partial \mathrm{B}, \text{ s.t. } v^q = x^H A_q(\alpha) x, \ q = 1, \ldots, r+c+s\} \ .$$

Make use of the set $\mathcal{P}_{r+c} \subset \mathbb{R}^{r+c+s}$ defined by

$$\mathcal{P}_{r+c} = \{v \in \mathbb{R}^{r+c+s} \ : \ v^q \geq 0, \ q = 1, \ldots, r+c; \ v^q = 0, \ q = r+c+1, \ldots, r+c+s\} \ .$$

Fact 1.3. [6] If $0 \in W(\alpha) + \mathcal{P}_{r+c}$ for some $\alpha \in \mathbb{R}$, then $0 \in W(\beta) + \mathcal{P}_{r+c}$ for all $\beta \leq \alpha$. □

Fact 1.4. [6]

$$\mu_{\mathcal{K}}(M) = \inf_{\alpha \geq 0} \{\sqrt{\alpha} \ : \ 0 \notin W(\alpha) + \mathcal{P}_{r+c}\} \ .$$

In terms of the distance function $c \ : \ \mathbb{R} \rightarrow \mathbb{R}^+$ given by

$$c(\alpha) = \min\{\|v\| \ : \ v \in W(\alpha) + \mathcal{P}_{r+c}\} \ ,$$

one have equivalently

$$\mu_{\mathcal{K}}(M) = \inf_{\alpha \geq 0} \{\sqrt{\alpha} \ : \ c(\alpha) > 0\} \ .$$

□

The above results mean that $\mu_{\mathcal{K}}(M)$ is the square root of α such that the set

$$\left\{ \begin{bmatrix} x^H A_1(\alpha) x \\ \vdots \\ x^H A_{r+c}(\alpha) x \end{bmatrix} \ : \ x^H A_i(\alpha) x = 0, \ i = r+c+1, \ldots, r+c+s, \ x \in \partial \mathrm{B} \right\}$$

first touches the negative orthant when α changes from $+\infty$ to 0. However, for $r = 0$ (no real uncertainty), $\mu_{\mathcal{K}}(M)$ can also be represented by [17]

$$\mu_{\mathcal{K}}(M) = \max_{\substack{\theta \in \mathbb{R} \\ x \in \partial \mathrm{B}}} \{\theta \ : \ \|P_q M x\| = \theta \|P_q x\|, \ q = 1, \ldots, c\} \ ,$$

or

$$\mu_{\mathcal{K}}(M) = \inf_{\alpha \geq 0}\{\sqrt{\alpha} \ : \ 0 \notin W(\beta) \text{ for all } \beta > \alpha\} \ .$$

Under this condition, $\mu_{\mathcal{K}}(M)$ is the square root of the α such that $W(\alpha)$ first touches the origin when α change from $+\infty$ to 0.

Algorithm 1.1. [4,6] (Computation of $\mu_{\mathcal{K}}(M)$)

Step 0. Set $\alpha_0 = \bar{\sigma}^2(M)$ and $k = 0$.

Step 1. Set $\alpha_{k+1} = \alpha_k - c(\alpha_k)$.

Step 2. Set $k = k + 1$ and go to Step 1.

□

It was proved in [4,6] that the sequence $\{\alpha_k\}$ generated by Algorithm 1.1 is monotone decreasing and

$$\lim_{k\to\infty} \sqrt{\alpha_k} = \mu_{\mathcal{K}}(M) \ .$$

Algorithms [1,18] exist to compute the distance from 0 to $\text{co}(W(\alpha) + \mathcal{P}_{r+c})$, where $\text{co}(W(\alpha) + \mathcal{P}_{r+c})$ is the convex hull of $W(\alpha) + \mathcal{P}_{r+c}$. When $W(\alpha)$ is convex, this yields $c(\alpha)$. This is the case whenever $r + c + s \leq 3$, i.e.. in the case of

1. Three or fewer complex blocks ($r = 0, \ c \leq 3$).

2. One real scalar block and one complex block ($r = 1, \ c = 1, \ s = 1$).

However for $r + c + s > 3$, $W(\alpha)$ may not be convex. The existing algorithm would then yield, instead of $c(\alpha)$, the value

$$c'(\alpha) = \min\{\|v\| \ : \ v \in \text{co}\ (W(\alpha) + \mathcal{P}_{r+c})\}$$

which is a lower bound for $c(\alpha)$. Using this value, instead of $c(\alpha)$, Algorithm 1.1 would then yield an upper bound $\mu'_{\mathcal{K}}(M)$ for $\mu_{\mathcal{K}}(M)$, with

$$\mu'_{\mathcal{K}}(M) = \inf_{\alpha \geq 0} \{ \sqrt{\alpha} \ : \ c'(\alpha) > 0 \} \ .$$

For $r = 0$ (no real uncertainty), it was shown in [4] that this new upper bound $\mu'_{\mathcal{K}}(M)$ is equal to $\inf_{D \in \mathcal{D}_{\mathcal{K}}} \bar{\sigma}(DMD^{-1})$. The following property is also proved in [4].

Property 1.5. [4] For $r = 0$ (no real uncertainty)

$$\mu_{\mathcal{K}}(M) = \inf_{D \in \mathcal{D}_{\mathcal{K}}} \bar{\sigma}(DMD^{-1})$$

if, and only if, there exists a vector $\lambda \in \mathbb{R}^c$, with $\lambda^i \geq 0$ for all i, such that the set $W(\mu^2_{\mathcal{K}}(M))$ is contained in the closed half space

$$H(\lambda) = \{ v \in \mathbb{R}^c \ : \ \langle v, \lambda \rangle \geq 0 \} \ .$$

$\square$

Since Fact 1.4 converts the evaluation of the structured singular value to the evaluation of the distance between the origin and a family of the set $W(\alpha) + \mathcal{P}_{r+c}$ in $\mathbb{R}^{r+c+s}$ space, a key question is 'Can we use the properties of m-form numerical range to help us in computing $c(\alpha)$?'. To explore this problem, we propose some algorithms to find the boundary of $W(\alpha)$.

CHAPTER
TWO

Graphically Displaying m-Form Numerical Range

2.1 Introduction

In this chapter, we investigate the question of graphically displaying the m-form numerical range W. Based on the results of the previous chapter, we propose several procedures to display any specified 2-dimensional or 3-dimensional section of the boundary of W. In Section 2, we first review an algorithm [4,8] to draw W for 2-dimensional case ($m = 2$). We point out a valuable property of this algorithm. In Section 3, we show that if one rotates and translates an m-form numerical range so as to map any k-dimensional section to the subspace spanned by the last k coordinate axes, the resulting set is still an m-form numerical range. In Section 4, we propose an algorithm to plot the 2-dimensional section of the *convex hull* of W. In Section 5, we investigate some properties of W for *non-convex* case. We show several interesting pictures of Doyle's example (see Appendix B) and one example of the structured singular value problem under parametric uncertainty. In Section 6, we propose an algorithm to draw the surface of the desired 3-dimensional section of an m-form numerical range using the NCAR [5] graphics software.

2.2 Two-Dimensional Case

In [4,8], a simple algorithm was proposed to plot the boundary of W when this set is in $\mathbb{R}^2$ ($m = 2$). In this section, after describing the algorithm, we show that the density of points plotted around any point is proportional to the curvature of the boundary at this point. In the following sections, we will generalize the algorithm to plot the boundaries of W for higher dimensional case.

Now consider the case when $W \subset \mathbb{R}^2$. Suppose W is strictly convex, i.e., for any $x,\ y \in W$,

$$\lambda x + (1-\lambda)y \in \mathrm{int}W \text{ for all } \lambda \in (0,1)$$

where $\mathrm{int}W$ denotes the interior points of W. Then clearly there is a one to one correspondence between the points of the boundary of W and the supporting hyperplanes to W, namely, for any $v \in \partial W$ (here we denote the the boundary of W by ∂W) there exists a unit vector $u = [\cos\theta\ \ \sin\theta]^{\mathrm{T}}$ such that v achieves the minimum in

$$\min\{\langle w,u\rangle \ :\ w \in W\}\ .$$

In view of the definition of W, $w^i = x^H A_i x$, for $i = 1,2$, where x achieves the minimum in

$$\min_{x\in\partial\mathbf{B}}\{x^H(\cos\theta A_1 + \sin\theta A_2)x\}\ ,$$

i.e., x is a unit eigenvector corresponding to the smallest eigenvalue of $\cos\theta A_1 + \sin\theta A_2$. This leads to the following procedure [4,8].

Procedure 2.1. [4,8]

Step 0. Set $\theta = 0$ and $N =$ a large integer.

Step 1. Let x be any unit eigenvector corresponding to the smallest eigenvalue of $\cos\theta A_1 + \sin\theta A_2$. Set

$$y_2 = \begin{bmatrix} x^H A_1 x \\ x^H A_2 x \end{bmatrix} .$$

If $\theta \neq 0$, draw the line segment $\overline{y_1 y_2}$. If $\theta \geq 2\pi$, stop.

Step 2. Set $y_1 = y_2$, $\theta = \theta + 2\pi/N$ and go to Step 1.

☐

Figure 2 depicts $W(A_1, A_2)$ with

$$A_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \quad A_2 = j \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} .$$

Here $W(A_1, A_2)$ consists of the convex hull of a disk of radius 1 center at $(-1, 0)$ and a point at $(1, 0)$ (this example is borrowed from [19], where a different algorithm is used to plot W). In agreement with Property (1.4), the point (1,0) is the image of an eigenvector $[1\ 0\ 0]^T$ of all linear combination of A_1 and A_2, i.e., an eigenvector of both A_1 and A_2. Figure 3 is a repeated plotting of the matrices given above by Procedure 2.1 except that in Step 2 a asterisk '$*$' is marked at the point y_2 and no line segment is drawn for each (y_1, y_2) pair. Same as that of Figure 3, Figure 4 depicts another $W(A_1, A_2)$ with arbitrarily generated A_1 and A_2 matrices. Figures 3 and 4 show that, while we uniformly divide 2π in Procedure 2.1 (Step 2), the corresponding points in the ∂W are not uniformly distributed and it seems that there are more points plotted around any point wherever the corresponding curvature is large. The following proposition shows that, in fact, the density of distribution at any point on ∂W is *proportional* to its curvature.

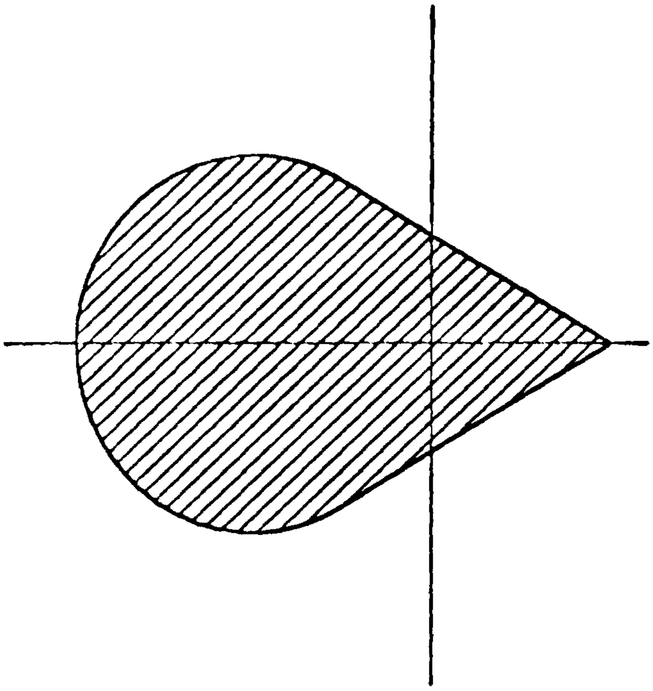

Figure 2: Example of Procedure 2.1

Proposition 2.1. Let $y_2(\theta)$ be the supporting point of hyperplane H with unit normal vector $u = [\cos\theta \ \sin\theta]^T$, as constructed in Step 1 of Procedure 2.1. Suppose that $y_2(\theta)$ is differentiable at θ. Then

$$\frac{dy_2(\theta)}{d\theta} = \frac{1}{K_\theta} \cdot T_\theta$$

where K_θ and T_θ are the curvature of ∂W at $y_2(\theta)$ and the unit tangent vector to ∂W at $y_2(\theta)$ respectively.

Proof. Let $s(\theta)$ represent the arc length of ∂W at $y_2(\theta)$. Then from the definition of curvature, since ∂W is smooth at θ, $\frac{1}{K_\theta} = \frac{ds(\theta)}{d\theta}$. Also, it is clear that $T_\theta = \frac{dy_2(\theta)}{ds(\theta)}$ is just the unit tangent vector at $y_2(\theta)$. Therefore

$$\frac{dy_2(\theta)}{d\theta} = \frac{dy_2(\theta)}{ds(\theta)} \cdot \frac{ds(\theta)}{d\theta}$$

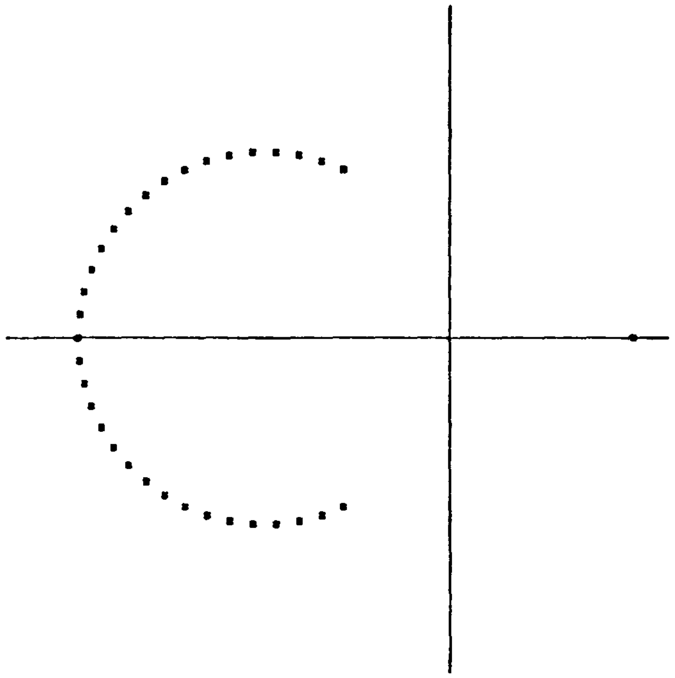

Figure 3: Generated Boundary Points of Figure 2

$$= \frac{1}{K_\theta} \cdot T_\theta$$

□

2.3 Coordinate Transformations

In this section, we show that if one rotates and translates an m-form numerical range so as to map any given k-dimensional section to the subspace spanned by the last k coordinate axes, the resulting set is still an m-form numerical range.

The intersection, denoted by Λ, of the m-form numerical range W of matrices $A_1, \ldots, A_m$ with a k-dimensional affine space can be expressed as

$$\Lambda = W \cap H_1 \cap \ldots \cap H_{m-k}$$

where, for $i = 1, \ldots, m-k$, H_i is the hyperplane defined as

$$H_i = \{v \in \mathbb{R}^m : \langle v, u_i \rangle = \lambda^i\}$$

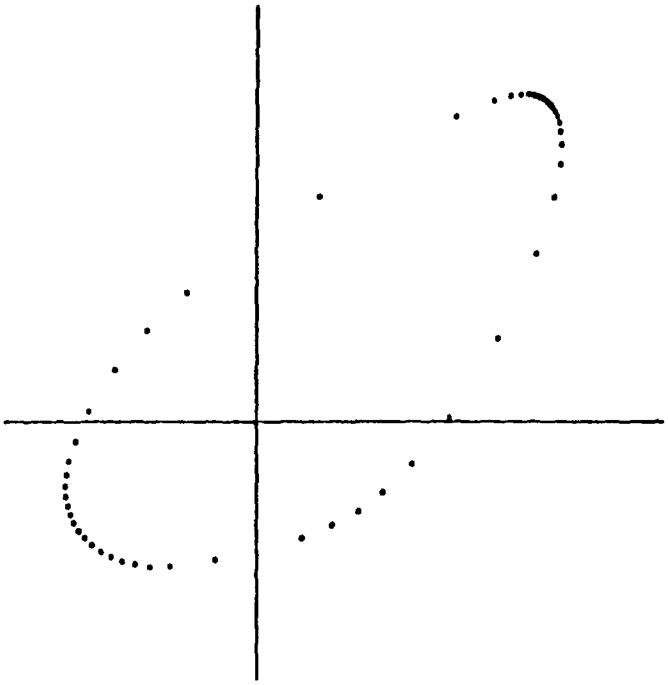

Figure 4: Property of Curvature

for some $u_i \in \mathbb{R}^m \backslash \{0\}$, $\lambda^i \in \mathbb{R}$. Without loss of generality, it can be assumed that the u_i's are orthonormal. Now consider the following coordinates transformation

$$\hat{v} = U^T v - \lambda \quad (\text{or } v = U(\hat{v} + \lambda)) \tag{2.1}$$

where

$$U = [u_1 \ldots u_m]$$

for some $u_{m-k+1}, \ldots, u_m$, such that U is an orthonormal matrix (i.e., $U^T U = I$), and

$$\lambda = [\lambda^1 \ldots \lambda^m]^T$$

with arbitrary $\lambda^{m-k+1}, \ldots, \lambda^m$. Transformation (2.1) is a composition of coordinates rotation and translation. Under this transformation, it is clear that the image of the k-dimensional section $\hat{\Lambda}$ of the m-form numerical range is

$$\hat{\Lambda} = \hat{W} \cap \hat{H}_1 \cap \ldots \cap \hat{H}_{m-k}$$

where $\hat{W}$ is the image of W and similarly for $\hat{H}_i$, $i = 1, \ldots, m-k$. Since

$$W = \{v \in \mathbb{R}^m \ : \ \exists x \in \partial \mathrm{B} \text{ s.t. } v^i = x^H A_i x \ \forall i \in \{1, \ldots, m\}\} \ ,$$

we obtain

$$\begin{aligned} \hat{W} &= \{\hat{v} \in \mathbb{R}^m \ : \ \exists x \in \partial \mathrm{B} \text{ s.t. } \hat{v} = U^T v - \lambda, \ v^i = x^H A_i x \ \forall i \in \{1, \ldots, m\}\} \\ &= \{\hat{v} \in \mathbb{R}^m \ : \ \exists x \in \partial \mathrm{B} \text{ s.t. } \hat{v}^i = x^H \hat{A}_i x, \ \forall i \in \{1, \ldots, m\}\} \\ &= W(\hat{A}_1, \ldots, \hat{A}_m) \end{aligned}$$

where, for $i = 1, \ldots, m$,

$$\hat{A}_i = \sum_{j=1}^{m} u_i^j A_j - \lambda^i I \tag{2.2}$$

Similarly, for $i = 1, \ldots, m-k$,

$$\begin{aligned} \hat{H}_i &= \{\hat{v} \in \mathbb{R}^m \ : \ \hat{v} = U^T v - \lambda, \ \langle v, u_i \rangle = \lambda^i\} \\ &= \{\hat{v} \in \mathbb{R}^m \ : \ \langle (\hat{v} + \lambda), U^T u_i \rangle \ = \lambda^i\} \\ &= \{\hat{v} \in \mathbb{R}^m \ : \ \hat{v}^i = 0\} \end{aligned}$$

Thus Λ can be obtained by translation and rotation from the intersection of the m-form numerical range of $\hat{A}_1, \ldots, \hat{A}_m$ with the span of the last k coordinate axes.

2.4 Two-Dimensional Sections of W

Consider now the question of displaying convex 2-dimensional sections of W ($m \geq 3$). In view of the discussion in Section 2.3, without loss of generality, suppose that the section of interest is defined by $w^i = 0, i = 1, \ldots, m-2$. The following procedure is suggested.

Procedure 2.2.

Step 0. Set $\theta = 0$ and $N =$ a large integer.

Step 1. Find $z^1, \ldots, z^{m-2} \in \mathbb{R}$ such that the Hermitian matrix

$$\sum_{i=1}^{m-2} z^i A_i + \cos\theta A_{m-1} + \sin\theta A_m$$

has a unit length eigenvector x corresponding to its smallest eigenvalue satisfying $x^H A_i x = 0$, $i = 1, \ldots, m-2$. Set

$$y_2 = \begin{bmatrix} x^H A_{m-1} x \\ x^H A_m x \end{bmatrix} .$$

If $\theta \neq 0$, draw the line segment $\overline{y_1 y_2}$. If $\theta \geq 2\pi$, stop.

Step 2. Set $y_1 = y_2$, $\theta = \theta + 2\pi/N$ and go to Step 1.

□

In view of Properties 1.1 and 1.2, Procedure 2.2 will correctly display the boundary of any 2-dimensional section of W when $m = 3$ and the 2-dimensional 'outer' boundary of W when $m = 4$ and the A_i's are real symmetric. It should be clear that, if the m-form numerical range W is not convex, Procedure 2.2 will display the boundary of the section of the *convex hull* of W. Figure 5 was generated by Procedure 2.2. It is the section defined by a 2-dimensional affine space passing through the origin of the convex hull of the m-form numerical range $W(\mu_{\mathcal{K}}^2(M))$ ($\mu_{\mathcal{K}}(M) \approx .87$) corresponding to an example due to Doyle [7], referenced to below as Example D (see Appendix B). Although Procedure 2.2 just displays a single 2-dimensional section of W, by its repeated use we can generate a family of convex 2-dimensional sections. Displaying all these curves may help us to better understand any 3-dimensional section of W, especially for $m = 3$. Furthermore, this will lead to a way to display 3-dimensional sections of W by

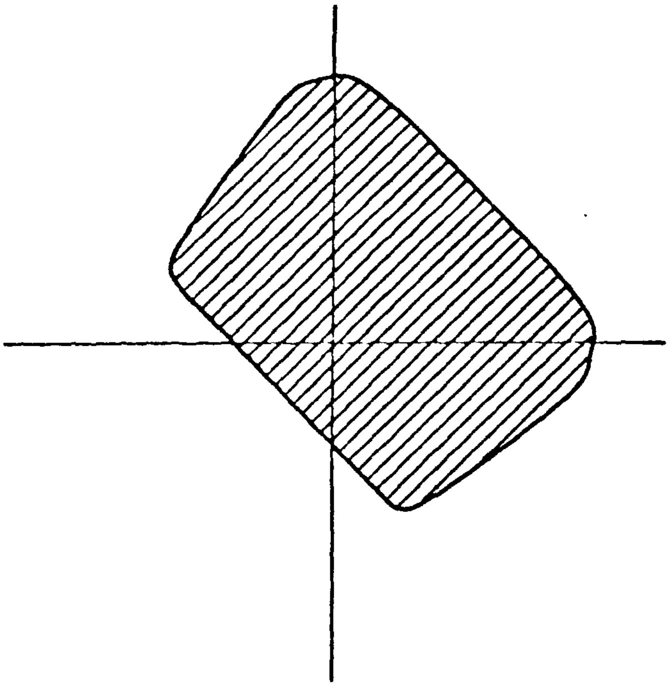

Figure 5: Convex Hull of Doyle's Example for $W(\mu_K^2(M))$

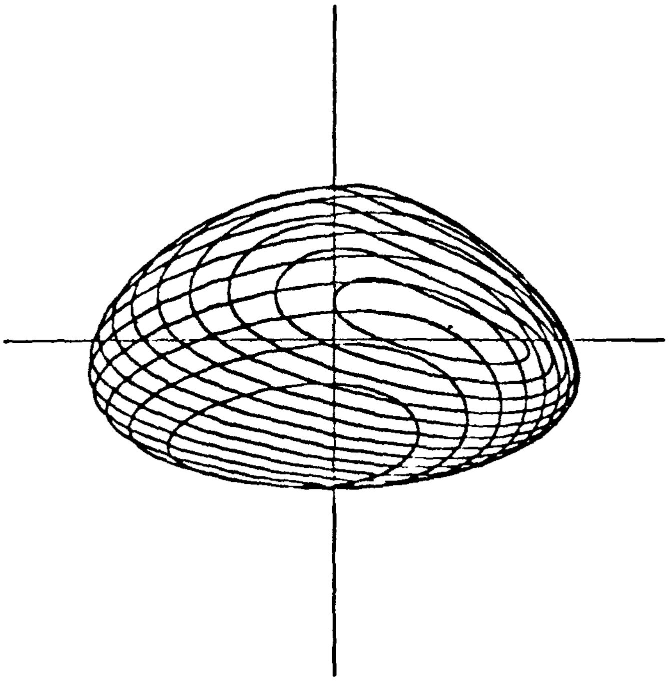

Figure 6: A Family of 2-Dimensional Curves

using the graphics package NCAR [5] (see Section 2.6 below). In the following procedure we denote the largest and smallest eigenvalues of a Hermitian matrix A by $\bar{\lambda}(A)$ and $\underline{\lambda}(A)$ respectively.

Procedure 2.3.

Step 0. Set $L =$ a large integer and $\nu = \underline{\lambda}(A_{m-2}) + (\bar{\lambda}(A_{m-2}) - \underline{\lambda}(A_{m-2}))/L$.

Step 1. Set $\theta = 0$ and $N =$ a large integer.

Step 2. Find $z^1, \dots, z^{m-2} \in \mathbb{R}$ such that the Hermitian matrix

$$\sum_{i=1}^{m-2} z^i A_i + \cos\theta A_{m-1} + \sin\theta A_m$$

has a unit length eigenvector x corresponding to its smallest eigenvalue satisfying for $i = 1, \dots, m-3$, $x^H A_i x = 0$, and $x^H A_{m-2} x = \nu$. Set

$$y_2 = \begin{bmatrix} x^H A_{m-1} x \\ x^H A_m x \end{bmatrix} .$$

If $\theta \neq 0$, draw the line segment $\overline{y_1 y_2}$. If $\theta \geq 2\pi$, go to Step 4.

Step 3. Set $y_1 = y_2$, $\theta = \theta + 2\pi/N$ and go to Step 2.

Step 4. Set $\nu = \nu + (\bar{\lambda}(A_{m-2}) - \underline{\lambda}(A_{m-2}))/L$. If $\nu \leq \bar{\lambda}(A_{m-2})$, go to Step 1. Stop otherwise.

$\square$

Figure 6 (generated by Procedure 2.3) depicts a family of 2-dimensional sections

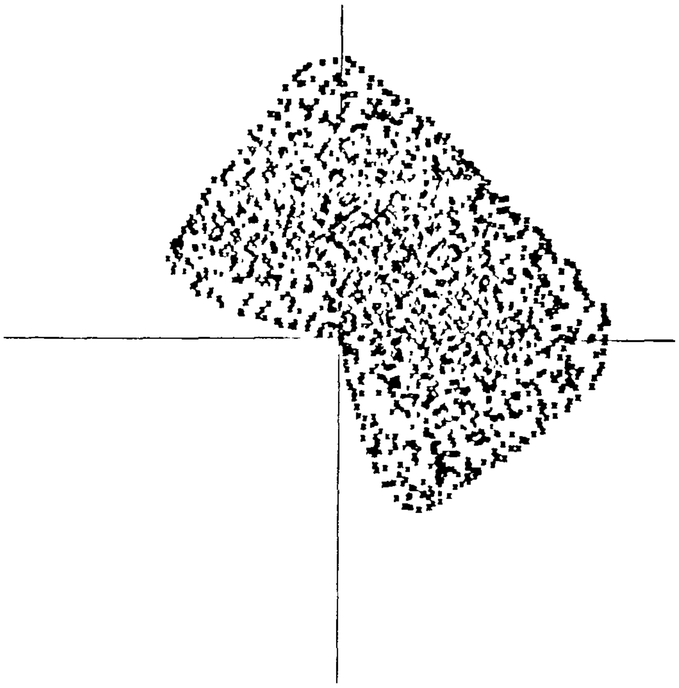

Figure 7: Doyle's Example for $W(\mu_K^2(M))$ by Brute Force

of the m-form numerical range $W(A_1, A_2, A_3)$ of arbitrarily generated matrices

$$
\begin{aligned}
A_1 &= \begin{bmatrix} 3+j0 & 0.5-j & 1-j.58 \\ 0.5+j1 & 2+j0 & 1+j12.9 \\ 1+j0.58 & 1-j12.9 & -2+j0 \end{bmatrix}, \\
A_2 &= \begin{bmatrix} 5+j0 & 2.37+j1 & 1+j2.789 \\ 2.37-j1 & 1+j0 & 5.23+j1 \\ 1-j2.789 & 5.23-j1 & -4.5+j0 \end{bmatrix}, \\
A_3 &= \begin{bmatrix} 2.5+j0 & 0.12-j2 & -5.67+j2 \\ 0.12+j2 & 1.5+j0 & 4.87-j2 \\ -5.67-j2 & 4.87+j2 & -5.3+j0 \end{bmatrix}.
\end{aligned}
$$

2.5 Non-convex boundary of W

In this section, we will investigate the geometric properties of W in the non-convex case. To investigate the question of graphically displaying non-convex 2-dimensional sections of W, we propose the following procedure as our first

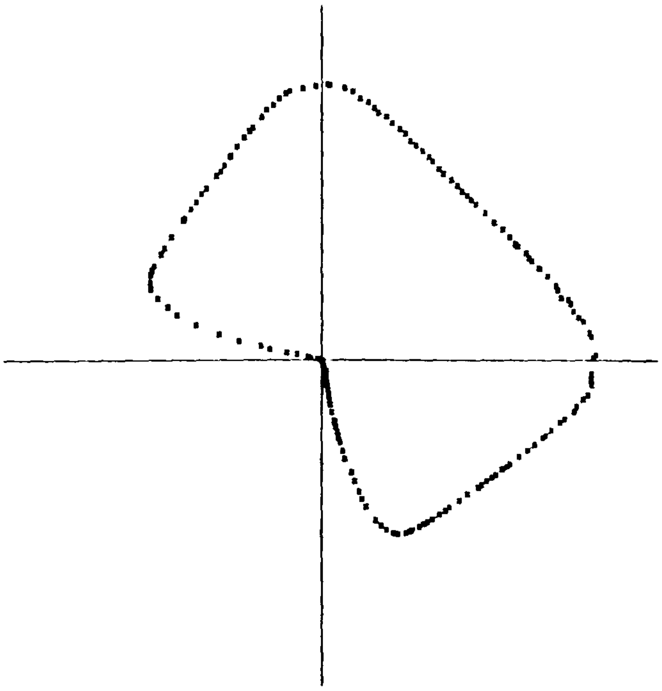

Figure 8: Boundary Points of Figure 7

step to solve this problem. This procedure uses 'brute force' to display the intersection of any given W with the span of the last two coordinate axes.

Procedure 2.4.

Step 1. Randomly generate $x_i \in \mathbb{C}^n$, $x_i \neq 0$, $i = 1, \ldots, m-1$.

Step 2. Find $\beta^i \in \mathbb{R}$ (e.g., using a Newton iteration), $i = 1, \ldots, m-2$ such that, for $j = 1, \ldots, m-2$,

$$(\sum_{i=1}^{m-2} \beta^i x_i + x_{m-1})^H A_j (\sum_{i=1}^{m-2} \beta^i x_i + x_{m-1}) = 0 \ .$$

Step 3. Set $x = \sum_{i=1}^{m-2} \beta^i x_i + x_{m-1}$. If $x = 0$, go to Step 1. Else set $x = x/\|x\|$ and

$$y = \begin{bmatrix} x^H A_{m-1} x \\ x^H A_m x \end{bmatrix} \ .$$

Draw a small dot at y. Go to Step 1.

□

In view of Property 1.5, Example D (see Appendix B) corresponds to a set $W(\mu_K^2(M))$ for which there is no $\lambda \geq 0$ such that $W(\mu_K^2(M))$ is contained in the closed half space $\{v \in \mathbb{R}^4 \ : \ \langle v, \lambda\rangle \ \geq 0\}$. Figure 7 (generated using Procedure 2.4), which displays that intersection of $W(\mu_K^2(M))$ with a 2-dimensional affine set passing through the origin, illustrates this point. Assume $\{b_1, \ldots, b_q\}$ is the set of the points generated by Procedure 2.4, the following procedure is proposed to find the boundary points of the set $\{b_1, \ldots, b_q\}$.

Procedure 2.5.

Step 0. Set N a large integer, $\hat{\theta}$ a small number, $\theta = 0$, and $c = (\sum_{i=1}^{q} b_i)/q$.

Step 1. set $u = [\cos\theta \ \ \sin\theta]^T$. Find $\hat{b}$ that solves

$$\max\left\{\|b - c\| \ : \ \frac{\langle b - c, u\rangle}{\|b - c\|} > \cos\hat{\theta}, \ b \in \{b_1, \ldots, b_q\}\right\} .$$

Draw a small dot at $\hat{b}$. If $\theta > 2\pi$, stop.

Step 2. $\theta = \theta + 2\pi/N$. Go to Step 1.

□

Figure 8 (generated by Procedure 2.5) depicts the boundary points of the set of points showed in Figure 7. Clearly Procedure 2.4 is computationally expensive. As an alternative, one can try to make use of Property 1.4. This possibility was investigated in the particular case of Example D. Figure 9 shows the exact boundary of the section displayed on Figure 7. The 'convex' portion of this boundary was drawn on Figure 5 using Procedure 2.2. It turns out that, in this case, the 'non-convex' portion is constituted of images of eigenvectors corresponding to the *second* largest eigenvalue of the linear combination $\sum_{k=i}^{m} w^i A_i$,

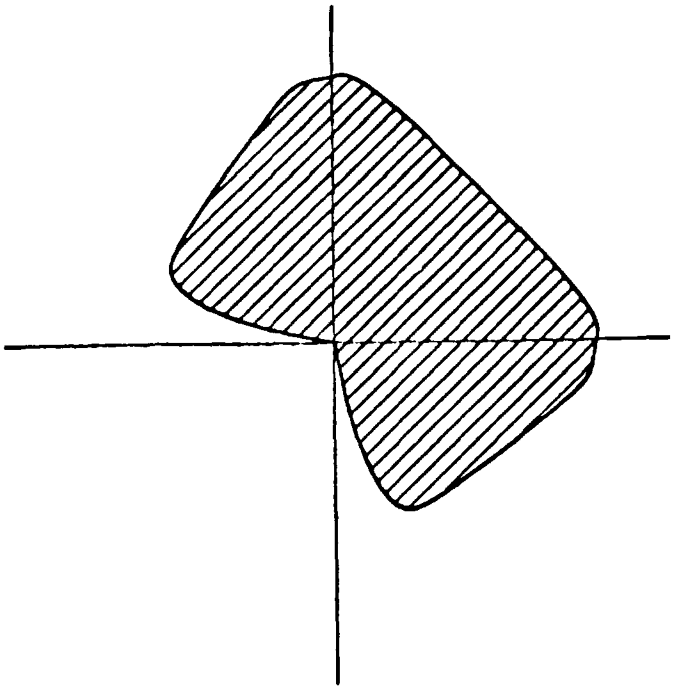

Figure 9: Non-Convex Section of Doyle's Example

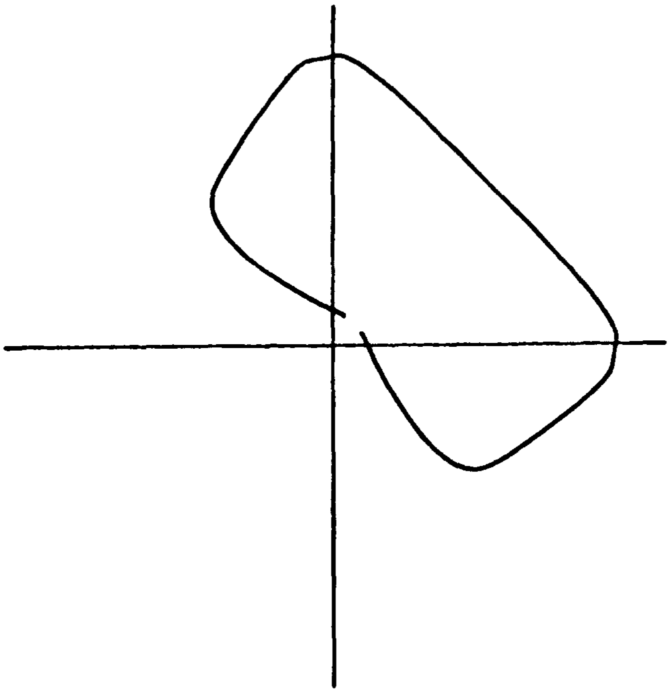

Figure 10:

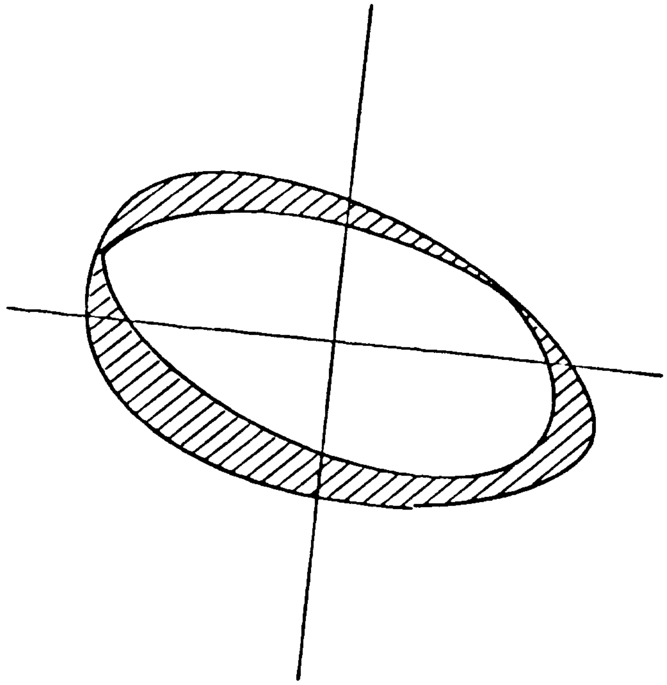

Figure 11:

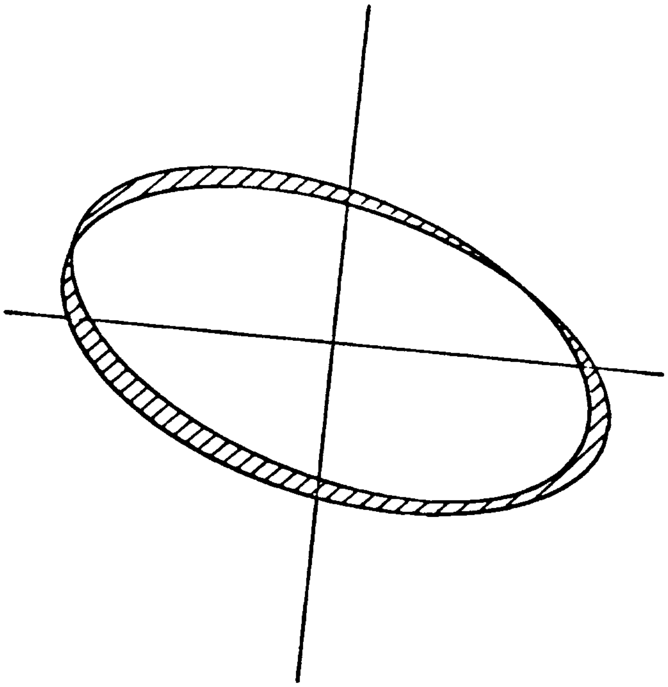

Figure 12:

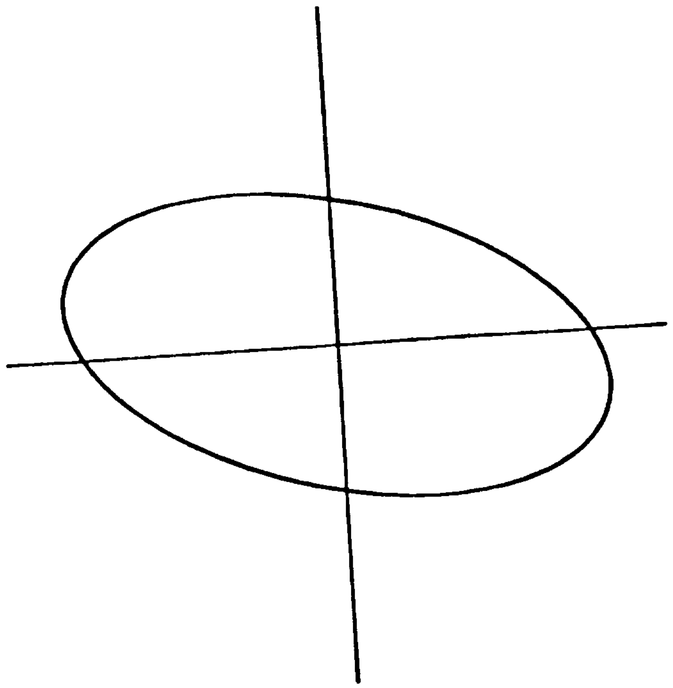

Figure 13:

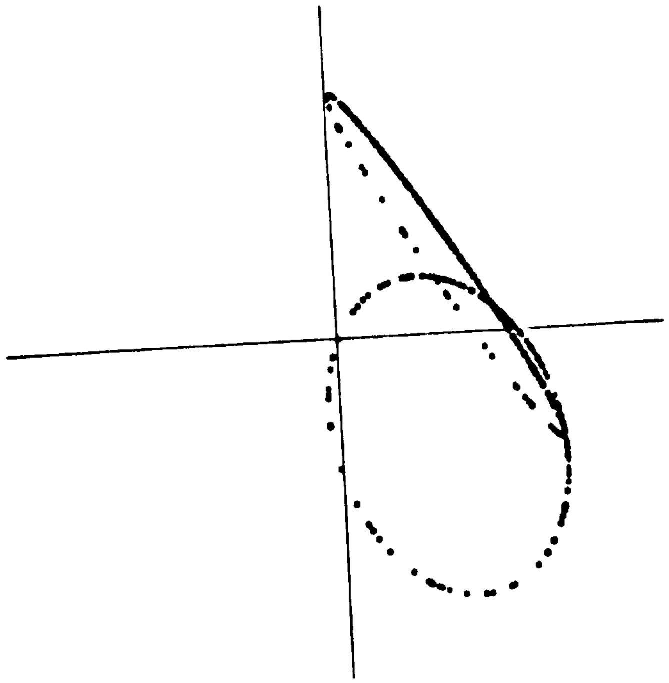

Figure 14:

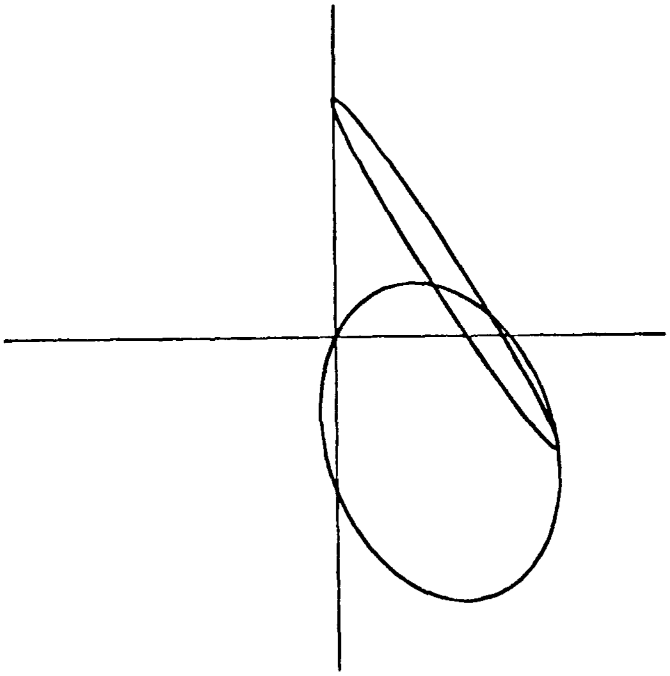

Figure 15:

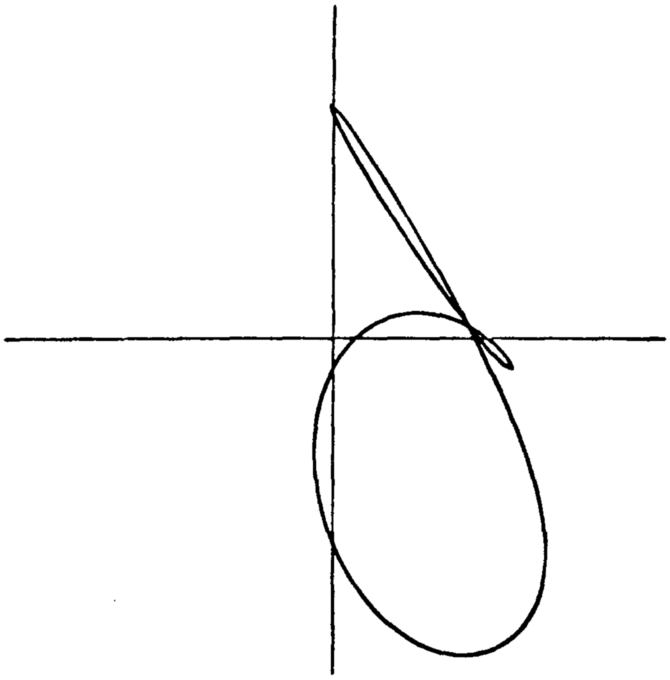

Figure 16:

with w a vector orthogonal to the hyperplane 'tangent to $W(\mu^2_{\mathcal{K}}(M))$' at the point under consideration. A similar technique was used to generate Figures 10 to 13. These also come from Example D, this time with $\alpha = 1$. As seen in Figure 10, co$W(1)$ touches the origin (while $W(1)$ does not) and the corresponding supporting hyperplane H intersects $W(1)$ at more than one point. Clearly, the 'contact surface' is not convex so that, in view of Properties 1.1 and 1.2, it must be an ellipsoid. This is confirmed by Figures 11 to 13 which depict 2-dimensional sections contained in 3-dimensional sections of W defined by hyperplanes parallel to H. The discontinuity in Figure 10 is due to two inflection points on the boundary. Consider the 2-dimensional case, the angle θ of the normal vector $u = [\cos\theta \sin\theta]^T$ of the 'tangent' hyperplane of the boundary point will not change monotonely around the inflection point. However, the corresponding θ defined in Procedures 2.1 and 2.2 increases monotonely. Under this conditin, Procedures 2.1 and 2.2 can't generate a continuous boundary curve. Figures 14 to 16 show 2-dimensional sections of $W(A_1, \ldots, A_6)$ which correspond to the structured singular value problem under parametric uncertainty with

$$M = \begin{bmatrix} 1 & 0 & -1 \\ 2 & -1 & 3 \\ 0 & 1 & -2 \end{bmatrix},$$

and the block-structure $\mathcal{K} = (1,1,1,1;)$. Then define

$$A_i = \alpha P_i - M^H P_i M, \text{ for } i = 1,2,3;$$

$$A_i = j(M^H P_i - P_i M), \text{ for } i = 4,5,6.$$

Figure 14 (generated by brute force method – Procedure 2.4) depicts the section

$$\left\{ \begin{bmatrix} x^H A_1 x \\ x^H A_2 x \end{bmatrix} : x^H A_i x = 0, \text{ for } i = 3, \ldots, 6, \text{ and } x \in \partial\mathrm{B} \right\}$$

of $W(A_1, \cdots, A_6)$ with $\alpha = 9$. Let

$$x = \begin{bmatrix} a_1 + jb_1 \\ a_2 + jb_2 \\ a_3 + jb_3 \end{bmatrix} \in \partial \mathrm{B} \ ,$$

and set

$$\begin{aligned} x^H A_3 x &= (5a_3 - a_2)(a_3 + a_2) + (5b_3 - b_2)(b_3 + b_2) = 0 \ ; \\ x^H A_4 x &= j(a_1 b_3 - a_3 b_1) = 0 \ ; \\ x^H A_5 x &= j2(a_1 b_2 - a_2 b_2) + j3(a_3 b_2 - a_2 b_3) = 0 \ ; \\ x^H A_6 x &= j(a_2 b_3 - a_3 b_2) = 0 \ . \end{aligned}$$

Then, we have

$$\begin{aligned} x^H A_1 x &= (4a_1 - a_3)(2a_1 + a_3) + (4b_1 - b_3)(2b_1 + b_3) \ ; \\ x^H A_2 x &= -(2a_1 - 7a_3)(2a_1 + a_3) - (2b_1 - 7b_3)(2b_1 + b_3) \ . \end{aligned}$$

By simplifying the above equations, we draw the exact curves of Figure 14 in Figure 15 which consist of two ellipses. Figure 16 shows another 2-dimensional section perturbed a little bit from that of Figure 15. These figures show that the set

$$\left\{ \begin{bmatrix} x^H A_1 x \\ x^H A_2 x \\ x^H A_3 x \end{bmatrix} \ : \ x^H A_i x = 0, \ i = 4, 5, 6, \ \text{and} \ x \in \partial \mathrm{B} \right\}$$

has no interior point.

2.6 Three-Dimensional Sections of W

In the previous sections, we discussed several procedures to display 2-dimensional section of a given m-form numerical range. We now present a procedure to display the surfaces of 3-dimensional sections of W $(m \geq 3)$. We will employ the graphics software NCAR [5].

The NCAR graphics software consists of graphics utilities such as multivariable graphs, contour plotting, 3-dimensional surface drawing and world map projection. These utilities perform graphics output using low-level graphics subroutines in NCAR system plot package. The output of the system plot package is a file of metacode, a device independent plot file which may be post-processed (interpreted) and plotted on any device for which a translator is available. To use the NCAR graphics utility, one must link with the NCAR library and call the desired graphics utility routine through a calling routine. The routine ISOSRF is one of 3-dimensional surface drawing utilities in NCAR. Suppose that given a function $r(x, y, z) : \mathbb{R}^3 \to \mathbb{R}$, one would like to draw the portion of the isosurface $r(x, y, z) = 0$ for which $x_1 \le x \le x_2$, $y_1 \le y \le y_2$ and $z_1 \le z \le z_2$. In order to use ISOSRF, one must construct a 3-dimensional array T (with size $n_i \times n_j \times n_k$, for some integers n_i, n_j and n_k) such that for any i, j and k

$$T(i,j,k) = r((\frac{x_2 - x_1}{n_i - 1})i + x_1,\ (\frac{y_2 - y_1}{n_j - 1})j + y_1,\ (\frac{z_2 - z_1}{n_k - 1})k + z_1)\ .$$

As mentioned in Section 2.4, Procedure 2.3 may be used to generate a family of convex 2-dimensional sections of the m-form numerical range. For the case when $m = 3$, Procedure 2.3 generates a set of boundary points of the m-form numerical range. Let $\{b_1, \ldots, b_q\}$ denote those boundary points. We now propose a function $r(x, y, z)$ such that the isosurface $r(x, y, z) = 0$ represents (more or less) the boundary points $\{b_1, \ldots, b_q\}$ of the m-form numerical range. Define (let $b \in \mathbb{R}^3$)

$$r(b) = \frac{\|b - c\|^2}{\|\hat{b} - c\|^2} - 1$$

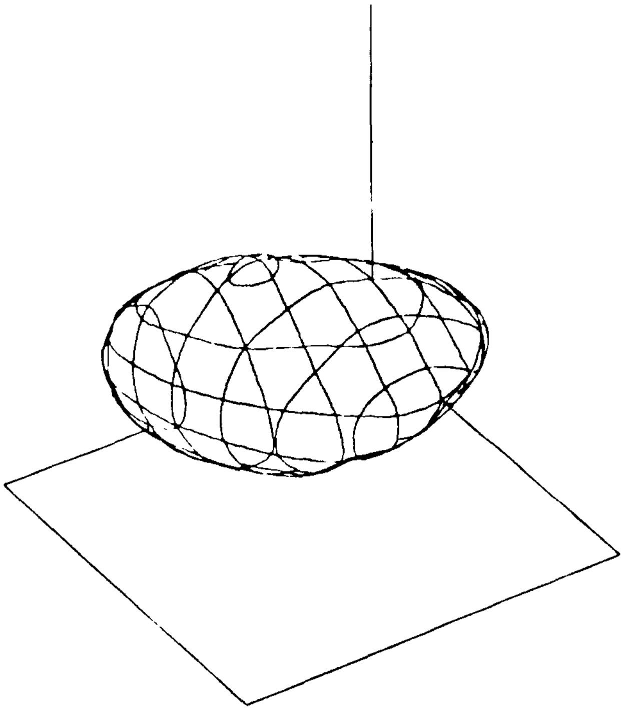

Figure 17: 3-Dimensional Convex Surface

where $c = \frac{\sum_{i=1}^{q} b_i}{q}$ and $\hat{b}$ solves ,

$$\max_{i\in\{1,\ldots,q\}} \frac{\langle b-c, b_i-c\rangle}{\|b-c\|\;\|b_i-c\|} .$$

Then it is easily checked that $r(b) > 0$ if $b \notin W$; $r(b) < 0$ if $b \in \operatorname{int} W$: and $r(b) \approx 0$ if $b \in \partial W$. Figure 17 depicts the 3-dimensional surface of the m-form numerical range showed in Figure 6 which is then represented by a family of 2-dimensional sections. Similar to Procedure 2.3, we can generate a family of 2-dimensional boundary curves by repeated use of Procedures 2.4 and 2.5. Figures 18 and 19 depict the non-convex surface of the intersection of Doyle's example (see Appendix B) with a 3-dimensional affine set passing through the origin for the $W(\mu_K^2(M))$ case.

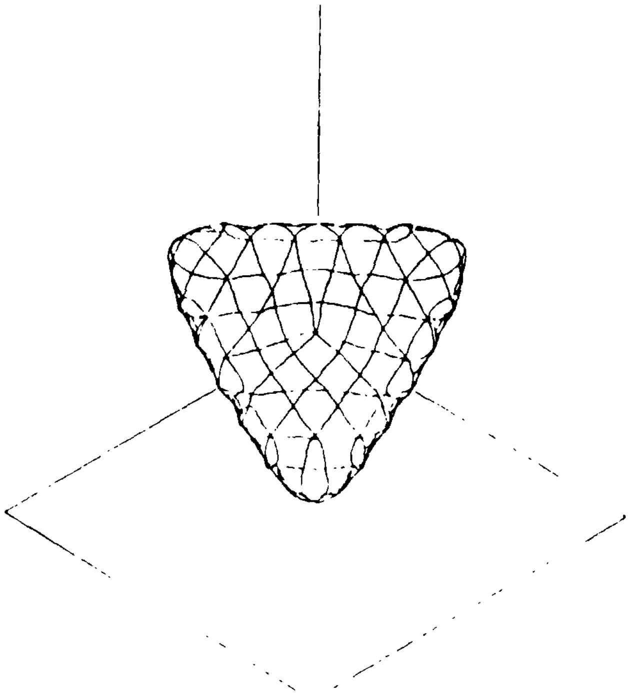

Figure 18: 3-Dimensional Non-Convex Surface of Doyle's Example

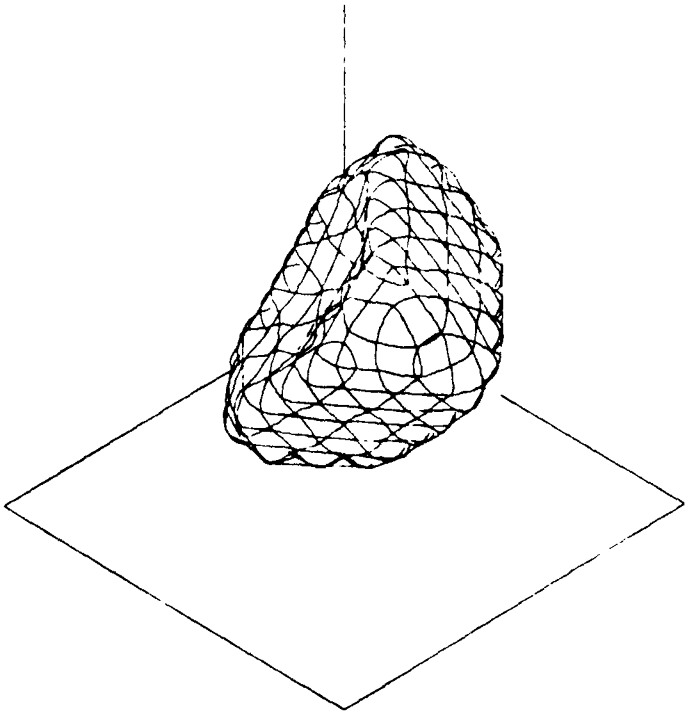

Figure 19: Different View Position of Figure 18

CHAPTER THREE

Conclusion

In Chapter 2, we proposed procedures to display the 2-dimensional sections of the m-form numerical range W. Procedure 2.2 will correctly display the 2-dimensional boundary of W when $m \leq 3$ and the 2-dimensional 'outer' boundary of W when $m = 4$ and A_i's are real. If the section under consideration is not convex, Procedure 2.2 will display the boundary of the 2-dimensional section of the convex hull of W. Procedure 2.4 uses brute force method to randomly generate the points that belong to the desired 2-dimensional section of W. We can display the non-convex 2-dimensional sections by this method even though it is computationally expensive. Procedure 2.5 sorts the points generated by Procedure 2.4 and finds the corresponding boundary points which could be used to approximate the exact boundary curve. In Figure 9, we show the non-convex boundary is the image of the unit eigenvector corresponding to the second largest eigenvalue of $\sum_{i=1}^{m} w^i A_i$, with w a vector orthogonal to the hyperplane tangent to $W(\mu_K^2(M))$ at the point under consideration. However, there is still no efficient algorithm to find the non-convex boundary of the m-form numerical range.

For graphically displaying the 3-dimensional sections of W, we employ the NCAR graphics software. Procedure 2.3 generates a family of 2-dimensional convex sections. Through the isosurface function $r(x, y, z)$ defined in Section 2.6 , we display the 3-dimensional surface of W by the NCAR graphics utility

– ISOSRF. By repeated use of Procedures 2.4 and 2.5, we can also generate a family of 2-dimensional sections and display the corresponding 3-dimensional surface by ISOSRF, especially for the non-convex case. It should be noted that not all kind of non-convex surface can be drawn through the isosurface function $r(x, y, z)$. For example, the non-convex surface of the 3-dimensional section corresponding to Figures 15 and 16 can not be drawn by using the isosurface function $r(x, y, z)$.

Appendix A

Software Package

A.1 Introduction

In this appendix we describe some software packages developed to display any desired 2-dimensional section of the m-form numerical range. According to Section 2.3, we develop program *mait* (short for *matrices* A_i*'s transformation*) to convert the structured singular value problem to the m-form numerical range and then transform the corresponding m-form numerical range to another m-form numerical range such that the desired 2-dimensional section is in the span of the last two coordinate axes (see Section 2.3). In order to find the desired 2-dimensional section of an m-form numerical range, program *nr* (short for *numerical range*) uses Procedures 2.2 and 2.3 to find the boundary points of the corresponding convex hull. In order to find the shape of the desired 2-dimensional section of the m-form numerical range, program *bf* (short for *brute force*) use Procedure 2.4 to randomly generate the points that belong to the desired 2-dimensional section. To find the boundary points of the set of points generated by *bf*, program *fbd* (short for *find boundary*) follows Procedure 2.5 to sort the points generated by *bf* and writes the boundary points in a output file. Programs *nr*, *bf* and *fbd* just generate the desired points and write them in a output file. To see the diagrams generated by programs *nr*, *bf*, and *fbd*, we have program *grap* (short for *graphics*) to draw the 2-dimensional diagrams on a specified device. To draw the 3-dimensional surface of a m-form numerical range, program *tisosrf* (short for *transformation for isosurface*) takes a set of boundary points as its input and generates a metacode file which could be post-processed by a translator *ncartrn* and be displayed on a specified device.

A.2 Program mait

According the following three equations, program *mait* converts structured singular value problem to the corresponding m-form numerical range and transforms the m-form numerical range to another m-form numerical range such that the desired 2-dimensional section is in the span of the last two coordinate axes.

Given an $n \times n$ complex matrix M and associated block-structure $\mathcal{K}$, we summarize the equations (1.6), (1.7), and (2.2) to achieve the above requirements. For $i = 1, \cdots r + c$

$$A_i = \alpha P_i - M^H P_i M \ . \tag{a.1}$$

For $i = r + c + 1, \cdots, r + c + s$

$$A_i = j(E_i M - M^H E_i) \ . \tag{a.2}$$

For $i = 1, \cdots, r + c$

$$\hat{A}_i = \sum_{j=1}^{m} u_i^j A_j - \lambda^i I \ . \tag{a.3}$$

where $E_i \in \mathcal{E}$ and $r + c$ is the number of block of uncertainties including parametric and dynamic uncertainties, and $U = [u_1 \ldots u_{r+c}]^T$ and λ are the rotation matrix and translation vector respectively. *mait* takes data from an input file - a file containing n, r (used to specify the number of block of parametric uncertainty), c (used to specify the number of block of dynamic uncertainty),

$(r+c)$-tuple block-structure $\mathcal{K}$, a $n \times n$ complex matrix M, the parameter α as that defined in $(a.1)$, 2 linear independent $(r+c) \times 1$ vectors $\bar{u}_{r+c-1}, \bar{u}_{r+c}$, and a $(r+c) \times 1$ translation vector $\bar{\lambda}$. By $(a.1)$ and $(a.2)$, the matrix M, the associated block-structure $\mathcal{K}$, and the parameter α specify the corresponding m-form numerical range. The vectors $\bar{u}_{r+c-1}$, $\bar{u}_{r+c}$, and $\bar{\lambda}$ together specify the desired 2-dimensional section which is the intersection of the corresponding m-form numerical range with the affine space $\{v \in \mathbb{R}^{r+c} \ : \ v \in \ \text{span}\{\bar{u}_{r+c-1}, \bar{u}_{r+c}\} + \bar{\lambda}\}$. To map the desired 2-dimensional section to the subspace spanned by the last two coordinate axes, we have the following procedure to generate the corresponding rotation matrix U and translation vector λ (see Section 2.3).

Procedure A.1.

Step 0. Generate a set of $(r+c) \times 1$ vectors $\{\bar{u}_1, \ldots, \bar{u}_{r+c-2}\}$ such that $\{\bar{u}_1, \cdots, \bar{u}_{r+c}\}$ is a linear independent set. Set $i = 1$ and $\bar{u}_0 = [0, \ldots, 0]^T$.

Step 1. Set

$$\begin{aligned} v_i &= \bar{u}_i - \sum_{j=1}^{i-1} \langle \bar{u}_i, u_j \rangle u_j \ ; \\ u_i &= \frac{v_i}{\|v_i\|} \ . \end{aligned}$$

Step 2. Set $i = i + 1$. If $i \leq r + c$, go to Step 1. Otherwise, set $U = [u_1, \cdots, u_{r+c}]^T$, $\lambda = U\bar{\lambda}$, and stop.

□

By $(a.1)$, $(a.2)$, and $(a.3)$, *mait* generates the corresponding A_i's and $\hat{A}_i$'s matrices and writes all the $n \times n$ Hermitian complex matrices $\hat{A}_i$'s into a file with the same name as that of the input file plus the extension name *.ai*.

The following is the format of the input file for program *mait*. One should follow the sequence of the parameters specified in the following table, otherwise there will be errors.

Note: The numbers at the end of each line are used for the purpose of explanation only, they do not actually appear in the input file.

$\mathtt{format\_mait}$ 1
$\mathtt{n}\ \mathtt{r}\ \mathtt{c}$ 2
$\mathbf{k}_1 \cdots \mathbf{k}_r\ \ \mathbf{k}_{r+1} \cdots \mathbf{k}_{r+c}$ 3
α 4
$\mathtt{m}_{11}\ \cdots\ \mathtt{m}_{1n}$
$\vdots$ 5
$\mathtt{m}_{n1}\ \cdots\ \mathtt{m}_{nn}$
$\bar{\mathtt{u}}^1_{r+c-1}\ \cdots\ \bar{\mathtt{u}}^{r+c}_{r+c-1}$
6
$\bar{\mathtt{u}}^1_{r+c}\ \cdots\ \bar{\mathtt{u}}^{r+c}_{r+c}$

$\bar{\lambda}^1 \cdots \bar{\lambda}^{r+c}$ 7

1. Parameter *format_mait* is the header of a input file. It should be specified at the very beginning of each input file for program *mait*.

2. Parameter n is used to specify the dimension of matrix M in $(a.1)$. The value of n must be strictly positive. Parameters r and c are used to specify the numbers of blocks of parametric uncertainty and dynamic uncertainty respectively.

3. Parameters $k_1, \cdots, k_{r+c}$ are used to specify the block-structure $\mathcal{K}$ of the given structured singular value problem.

4. Parameter α is used to specify the α in $(a.1)$.

5. Data $m_{11}, \cdots, m_{nn}$ are used to specify the $n \times n$ complex matrix M in $(a.1)$. The entries of matrix M are stored by the following row order:

$$m_{11}\ m_{12} \cdots m_{1n}\ m_{21} \cdots m_{nn}.$$

All $\tilde{m}_{ij}$'s are complex numbers . Each complex number must has its real part followed by its imaginary part. Real part and imaginary part must be separated by at least one blank space.

Note: The above representations for complex matrix and complex numbers will be followed throughout this appendix without any further mention.

6. Data $\bar{u}_i^1, \cdots, \bar{u}_i^{r+c}$, for $i = r+c-1, r+c$, are used to specify the desired 2 linear independent vectors $\bar{u}_{r+c-1}$ and $\bar{u}_{r+c}$.

7. Data $\bar{\lambda}^1, \cdots, \bar{\lambda}^{r+c}$ are used to specify the $(r+c) \times 1$ real translation vector $\bar{\lambda}$. The vectors $\bar{u}_{r+c-1}$, and $\bar{u}_{r+c}$ together with the vector $\bar{\lambda}$ specify the desired 2-dimensional section of the corresponding $r+c+s$-form numerical range.

After reading all the required data from the input file, *mait* generates all $\hat{A}_i$'s matrices by $(a.3)$ and writes them in a output file. The following is the format of the output file of program *mait*.

```
format_ai                     1
n m                           2
a11  ...  a1n
      :                       3
an1  ...  ann
:
a11  ...  a1n
      :                       m+2
an1  ...  ann
```

1. Parameter *format_ai* is the header for the output file of *mait*.

2. Parameter n and m are to specify the dimension and the number of Hermitian complex matrix stored in this file.

3. to $m+2$. All $n \times n$ Hermitian complex matrices $\hat{A}_i$, for $i = 1, \cdots, m$, are sequentially specified by these complex numbers a_{ij}'s.

The output file of *mait* could be used as the input file of programs *nr* and *bf* directly (see Sections A.3 and A.4 below). Let us assume the input file is *db*. To invoke *mait*, one just types

```
mait db
```

A message similar to the following appears on the terminal.

```
welcome to mait
reading db ...
computing Ai's ...
transforming Ai's ...
writing the transformed Ai's to db.ai ...
```

The bottom three lines correspond to three successive operations performed by *mait*. Each of these lines appears on the screen at the start of the corresponding operation. If any error is detected, *mait* reports the corresponding error message and aborts.

A.3 Program nr

The purpose of program *nr* (short for numerical range) is used to generate the boundary points of the intersection or projection of the m-form numerical range with the span of the any two of m coordinate axes. *nr* takes all A_i matrices from an input file which has the same format specified by the output file of *mait*. By following Procedure 2.2 or 2.3, *nr* generates the boundary points of convex hull of the corresponding intersection. All the generated boundary points are written to a output file which has the same name as that of the input file plus the extension name *.nr*. The syntax of *nr* is

```
nr [option]  input_file
```

where [option] contains any option specified by the user. The following options are interpreted by *nr*:

-e Have *nr* echo all the data in the input file on the screen.

-i *i j* Have *nr* choose A_i and A_j matrices as the A_{m-1} and A_m matrices in the Procedure 2.2 or Procedure 2.3. If **-s** (see the next option) option is not specified, *nr* follows the Procedure 2.2 to find the boundary points. Otherwise *nr* follows the Procedure 2.3. The default values of i and j are $m-1$ and m.

-s *k ns* Have *nr* choose A_k as the A_{m-2} matrix and *ns* as the L in Procedure 2.3 respectively. *nr* follows Procedure 2.3 (note A_{m-1}, A_m matrices are specified by A_i, and A_j matrices) to find the corresponding boundary points.

-p Specify the desired boundary points are the convex hull of the projection of the m-form numerical range to the span of the ith and jth coordinate axes.

-n *np* Specify the number of points per intersection to be generated. The default value of *np* is 50.

Let us assume the input file is *db*, the number of A_i's matrices is greater than 6, and each point generated by *nr* has its vector representation as $[x1\ x2\ \ldots\ xm]^T$. The following is the format of the output file of *nr*:

```
format_graph        1
m i j k             2
x1 x2 ... xm        3
```

```
y1 y2 ... ym           3
.
.
.
z1 z2 ... zm           3
0.12345  56789         4
```

1. Parameter *format_graph* is the header of the output file of *nr*. This header is used for programs *fbd*, *grap*, and *tisosrf* (see sections A.4, A.5, and A.6 below) to identify the format of this file.

2. Parameter m is used to specify the dimension of each point generated by *nr*. Parameters i , j and k are used to specify the data in this file are generated by setting A_i, A_j, and A_k matrices as the A_{m-1} , A_m, and A_{m-2} matrices. If the option **-s** is not specified when *nr* is invoked, then the value of k will be 0. Programs *fbd*, *grap*, and *tisosrf* use these three parameters to read the the ith, jth and kth coordinates of each point of the input file.

3. Data from $x1$ $x2 \ldots xm$ to $z1$ $z2 \ldots zm$ are the coordinates of the points generated by *nr*. Each set of m real numbers is the vector representation of a point in m-dimensional space.

4. Special codes 0.12345 56789 are used to separate the data of different 2-dimensional sections. These special codes are automatically appended to the end of the data of each section. Program *grap* (see section A.5 below) takes the output file of *nr* as its input file and uses these special codes to separate the data of different boundary curves. By this method, we could keep several boundary curves in the same file.

The following is an example to invoke *nr*.

```
nr -i 3 5 -s 2 60 -n 100 db
```

In the above example, *nr* will follow Procedure 2.3 to find boundary points, choose A_2, A_3, and A_5 matrices as the corresponding A_{m-2}, A_{m-1}, and A_m matrices in Procedure 2.3, generate 60 2-dimensional sections with 100 boundary points per section, and write the generated data in the output file *db.nr*. If any error is detected, *nr* reports the corresponding error message and aborts.

A.4 Programs bf and fbd

Similar to *nr*, program *bf* also displays the intersection of a m-form numerical range with the span of any two of its m coordinate axes. Instead of finding the boundary curves, *bf* randomly generates those points that belong to the desired 2-dimensional section of the given m-form numerical range by Procedure 2.4. The formats of the input and output files of *bf* are the same as that of *nr*'s except that the output file of *bf* has the extension name *.bf.* The syntax of *bf* is

```
bf [option] input_file
```

where [option] contains any option specified by the user. The following options are interpreted by *bf* :

-e Have *bf* echo all the data in the input file on the screen.

-i *i j* Have *bf* choose A_i and A_j matrices as the A_{m-1} and A_m matrices in Procedure 2.4. The default values of i and j are $m-1$ and m.

-n *nop* Specify the number of points to be generated. The default value of *nop* is 5000.

Because relatively *bf* need much more computation time to have a good enough approximation of the desired section, we omit the optin **-s** specified in the syntax of *nr*. To get a family of 2-dimensional sections, one need to invoke *bf* for each section. This also can be done by invoking *bf* at different computers. Let us assume the input file is *db* and the number of A_i's matrices is greater than 5. The following is an example to invoke *bf*.

```
bf -i 4 5 -n 5000 db
```

In the above example, *bf* will select the A_4 and A_5 matrices as the A_{m-1} and A_m matrices in the Procedure 2.4 respectively. The number of points to be generated is 5000. All these points are written in the file *db.bf* with the same format as that specified by the output file of *nr*. If any error is detected, *bf* reports the corresponding error message and aborts.

Program *fbd* follows Procedure 2.5 to find the boundary points of the set of points generated by program *bf*. It takes data from the output file of *bf* and writes the boundary points in a file with the same name as that of input file plus the extension name *.bd*. The format of the output file of *fbd* is the same as that of *nr*'s. Assume the input file is *pd.bf*, the following is an example to invoke *fbd*.

```
fbd -n 300 pd.bf
```

The option **-n** is the only option of *fbd*. This option specifies the number of boundary points generated by *fbd*. The default value of this number is 200. In the above example, we specify the number of boundary points is 300. Once complete, an output file *dp.bf.bd* will be generated.

A.5 Program grap

grap is a program for plotting 2-dimensional curves or making a small mark at the corresponding vector position for each point on a specified device which could be a graphics terminal or a imagen printer. It takes the output files of programs *nr*, *bf*, and *fbd* as its input file and plots the desired diagram on a specified device. According to the format of the input file, each point is represented as an m-dimensional vector. Since *grap* just plots 2-dimensional diagram, it only uses two (these are specified by parameters i and j in the input file) of m coordinates of each point to plot the diagram. This means *grap* plots 2-dimensional projection diagram from a higher dimensional space. Throughout this section, we only use the ith and jth coordinates to identify a m-dimensional point. For simplicity, we assume the graphics screen is a 2-dimensional plane, There are three kinds of graphics modes in *grap*. The first one only draws a small mark at the corresponding screen position for each point. The second one draws a line segment between two corresponding screen positions for every two successive points stored in the input file. Combining previous two modes together, the last one displays the diagram by drawing small marks at each corresponding screen positions and connecting a line segment between two successive points on the screen. It is possible that a input file may have more than one diagram to be plotted. Under this condition, *grap* use a set of special codes to separate each group of data (see Section A.3 for detail description). It should note that *grap* accepts more than one input file. By this method, *grap* takes different diagrams from different files and plots them on the same screen. The syntax of *grap* is

```
grap [option] input_file1 input_file2 ... input_filen
```

The following options are interpreated by *grap*:

-e Have *grap* echo the parameters read from the input file except the coordinate of each point.

-h Have *grap* choose HP2623 graphics terminal as the output device.

-t Have *grap* choose V550 tektronix emulating graphics terminal as the output device.

-imp Have *grap* choose imagen printer as the output device.

-m Have *grap* draw a small mark at the corresponding screen position for each point stored in the input file. This option is typically useful to display the diagram generated by program *bf.*

-c Have *grap* draw a line segment between two corresponding screen positions for every two successive points in the input file. This means *grap* draw a boundary curve by connecting every two successive points by a line segment. It should note that this is not suitable to display the diagram generated by program *bf* but is useful to draw the diagram generated by program *nr*.

-b Have *grap* choose both -m and -c options.

-w *wx1 wy1 wx2 wy2* Have *grap* define the coordinates of the left bottom corner and right upper corner of the graphics device as $(wx1, wy1)$ and $(wx2, wy2)$. All the points with the coordinate beyond these ranges will not appear on the graphics device. If this option is not specified, *grap* will automatically choose a set of suitable ranges such that all the points in the input file will be displayed on the graphics device..

Let us assume the input files are *db1* and *db2*, the graphic terminal is HP2623, the world coordinates are (-10 -10 10 10). To invoke *grap*, one just type:

```
grap -e  -h -c  -w -10 -10 10 10 db1 db2
```

In the above example, *grap* takes *db1* and *db2* as its input files, echo all the specified parameters for each input file, displays the boundary curves on a HP2623 terminal. All the points with coordinate beyond the ranges (-10 -10) and (10,10) will not be displayed.

A.6 Program tisosrf

By calling the NCAR graphics utility – ISOSRF, program *tisosrf* generates the corresponding 3-dimensional surface metacode from a family of 2-dimensional sections. The format of the input file of *tisosrf* is the same as that of program *grap*'s. In order to have a correct 3-dimensional surface diagram generated by *tisosrf*, we should have enough number of data stored in the input file. The output of *tisosrf* is a file named *metacode* which is a device independent plot file. Assume the input file is *bd*, the following is an example to invoke *tisosrf*.

```
tisosrf db
```

Once complete, a *metacode* file will be generated. This file could be plotted by the translator program *ncartrn* on HP2623, or Visual 550 graphics terminals. The syntax of *ncartrn* is

```
ncartrn [terminal] [metacode_file]
```

where **terminal** option could be -h or -t which are used to specify the output device as HP2623 graphics terminal or V550 tektronix emulating graphics terminal respectively. The default metacode file is the local file *metacode*, but one can specify any file that contains the metacode. Assume the file *bd* contains the desired metacode file, the following is an example to invoke *ncartrn*.

```
ncartrn -h bd
```

In the above example, *ncartrn* takes metacode file from file *bd* and displays the diagram on a HP2623 terminal.

Appendix B

Doyle's Example

For ease of reference, we reproduce here an example due to Doyle [7] where $r = 0$, $c = 4$, and equality in (1.5) does not hold. Some of background from [1] is given first. All the following results are restricted to the case $r = 0$ (no real uncertainty).

For a complex $n \times n$ matrix M, consider the singular value decomposition as

$$M = [U_1 \; U_2] \sum [V_1 \; V_2]^H ,$$

where $U_1, V_1 \in \mathbb{C}^{n \times q}$ with q the multiplicity of $\bar{\sigma}(M)$ and $[U_1 \; U_2]$, $[V_1 \; V_2]$ are unitary. Define

$$\nabla_2 = \{z \in \mathbb{R}^{c-1} \; : \; \exists v \in \mathbb{C}^q, \; \|v\| = 1 \text{ s.t. } z^i = v^H H_i v, \; \forall i \in \{1, \ldots, c-1\}\}$$

where

$$H_i = \frac{1}{2}(U_1^H M_i V_1 + V_1^H M_i^H U_1)$$

and

$$M_i = P_i M - M P_i \; .$$

The following two facts are proved in [1].

Fact B.1. $\inf_{D \in \mathcal{D}_\mathcal{K}} \bar{\sigma}(M) = \bar{\sigma}(M)$ if, and only if, $0 \in \mathrm{co}\nabla_2$.

Fact B.2. $\mu_\mathcal{K}(M) = \bar{\sigma}(M)$ if, and only if, $0 \in \nabla_2$.

Doyle's example is as follows. Let

$$M = \begin{bmatrix} a & 0 \\ ab & ab \\ ab & abj \\ \sqrt{1-2a^2} & \frac{-a^2(1+j)}{2\sqrt{1-2a^2}} \end{bmatrix} \begin{bmatrix} 0 & a \\ ab & -ab \\ ab & -abj \\ \frac{-a^2(1-j)}{2\sqrt{1-2a^2}} & \sqrt{1-2a^2} \end{bmatrix}^H$$

and

$$\mathcal{K} = (;1,1,1,1)$$

where $a = \sqrt{1 - \frac{\sqrt{3}}{3}}$, $b = \frac{\sqrt{2}}{2}$ and $j = \sqrt{-1}$. One obtains

$$H_1 = a^2 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \; H_2 = a^2 \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \; H_3 = a^2 \begin{bmatrix} 0 & j \\ -j & 0 \end{bmatrix}$$

and it is easy to check that ∇_2 is a circle with radius a^2 centered at the origin. Thus, by Facts B.1 and B.2, we have

$$\inf_{D \in \mathcal{D}_{\mathcal{K}}} \bar{\sigma}(DMD^{-1}) = \bar{\sigma}(M) \neq \mu_{\mathcal{K}}(M) \; .$$

References

[1] J.C. Doyle, "Analysis of Feedback Systems with Structured Uncertainties," *Proc. IEE-D* 129 (1982), 242–250.

[2] J.C. Doyle, J.E. Wall & G. Stein, "Performance and Robustness Analysis for Structured Uncertainty," *Proc. 21st IEEE Conf. on Decision and Control*, Orlando, Florida (December 1982).

[3] J.C. Doyle, "Structured Uncertainty in Control System Design," *Proceedings of the 24th IEEE Conference on Decision and Control*, Fort Lauderdale, Florida (December 1985).

[4] M.K.H. Fan & A.L. Tits, "m-Form Numerical Range and the Computation of the Structured Singular Value," *IEEE Trans. Automat. Control* AC-33 (1988).

[5] The National Center for Atmospheric Research, "The Introduction to The NCAR Graphic Software," *NCAR Technical Note* (January 1981).

[6] M.K.H. Fan, A.L. Tits & J. C. Doyle, "Robustness of Control Systems Under Joint Parametric and Dynamic Uncertainty," *Technical Research Report of Systems Research Center, University of Maryland, College Park, Maryland* (1987).

[7] J.C. Doyle, private communication, 1984.

[8] M.K.H. Fan, "Characterization and Computation of the Structured Singular Value," University of Maryland, PhD Thesis, College Park, MD 20742, 1986.

[9] O. Toeplitz, "Das algebraische Analogon zu einem Satze von Fejér," *Math. Z.* 2 (1918), 187–197.

[10] F. Hausdorff, "Der Wertvorrat einer Bilinearform," *Math. Z.* 3 (1919), 314–316.

[11] W.F. Donoghue, "On the Numerical Range of a Bounded Operator," *Michigan Math. J.* 4 (1957), 261–263.

[12] L. Brickman, "On the Field of Values of a Matrix," *Proc. Amer. Math. Soc.* 12 (1961), 61–66.

[13] Y.-H. Au-Yeung & Y.-T. Poon, "A Remark on the Convexity and Positive Definiteness Concerning Hermitian matrices," *Southeast Asian Bull. Math.* 3 (1979), 85–92.

[14] M.K.H. Fan & A.L. Tits, "On the Generalized Numerical Range," *Linear and Multilinear Algebra* 21 (1987).

[15] S. Friedland & R. Loewy, "Subspaces of Symmetric Matrices Containing Matrices with a Multiple First Eigenvalue," *Pacific J. Math.* 62 (1976), 389–399.

[16] Y.-H. Au-Yeung & N.-K. Tsing, "An Extension of the Hausdorff-Toeplitz Theorem on the Numerical Range," *Proc. Amer. Math. Soc.* 89 (1983), 215–218.

[17] M.K.H. Fan & A.L. Tits, "Geometric Aspects in the Computation of the Structured Singular Value," *Proceedings of the American Control Conference*, Seattle, Washington (June, 1986).

[18] John E. Hauser, "Proximity Algorithms: Theory and Implementation," Electronics Research Laboratory, University of California, Memo No. UCB/ERL M86/53, Berkeley, California, May 1986.

[19] M. Marcus & C. Pesce, "Computer Generated Numerical Ranges and Some Resulting Theorems," *Linear and Multilinear Algebra* 20 (1987), 121–157.

CURRICULUM VITAE

Name: Jia-Chang Wang.

Permanant Address: 4804 Osage St. College Park, MD 20740.

Degree and Date to be Conferred: Master of Science, 1988.

Place of Birth: Taipei, Taiwan, Republic of China.

Secondary Eduation: Chein Kuo High School, Taipei, Taiwan R.O.C..

Collegiate institution:	Dates	Degree	Date of Degree
University of Maryland, College Park, Maryland	9/85 - 5/88	M.S.	5/88
National Chiao Tung University, Taiwan R.O.C.	9/78 - 6/82	B.S.	6/82

Major: Control in Electrical Engineering.