Singular values of two-parameter matrices: An algorithm to accurately find their intersections [⋆]

Luca Dieci

*School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332
U.S.A.*

Alessandro Pugliese

*School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332
U.S.A. –and– Dip. di Matematica, Universitá di Bari, Via Orabona 4, 70125 Italy*

Abstract

Consider the Singular Value Decomposition (SVD) of a two-parameter function $A(x)$, $x \in \Omega \subset \mathbb{R}^2$, where Ω is simply connected and compact, with boundary Γ. No matter how differentiable the function A is (even analytic), in general the singular values lose all smoothness at points where they coalesce. In this work, we propose and implement algorithms which locate points in Ω where the singular values coalesce. Our algorithms are based on the interplay between coalescing singular values in Ω, and the periodicity of the SVD-factors as one completes a loop along Γ.

Key words: Two–parameter continuation, conical intersection, singular value decomposition, bisection, periodicity
1991 MSC: 15A18, 15A23, 65P30

Notation. An $(n{\times}n)$ real matrix is indicated with $A \in \mathbb{R}^{n\times n}$. We consider always the 2-norm of vectors and the induced norm for matrices. We adopt column partitioning notation; e.g., $A_{1:p,1:q}$ is the leading $(p \times q)$ submatrix of A, while $A_{:,1:q}$ are the first q columns of A. A matrix valued function $A : \mathbb{R} \to \mathbb{R}^{n\times n}$, continuous with its first l derivatives $(l \geq 0)$, is indicated as $A \in \mathcal{C}^l(\mathbb{R}, \mathbb{R}^{n\times n})$, and similarly if A is a function of two real variables, $A \in \mathcal{C}^l(\mathbb{R}^2, \mathbb{R}^{n\times n})$. If $l = 0$, we also simply write

[⋆] This work was supported in part under NSF DMS 0139895, INDAM GNCS. The authors thank Christian Lubich and Alessandro Spadoni for pointing out relevant references.

Email addresses: dieci@math.gatech.edu (Luca Dieci), pugliese@math.gatech.edu (Alessandro Pugliese).

$A \in \mathcal{C}$. If $A \in \mathcal{C}^l(\mathbb{R}, \mathbb{R}^{n\times n})$ is periodic of (minimal) period $\tau > 0$, we write it as $A \in \mathcal{C}^l_\tau(\mathbb{R}, \mathbb{R}^{n\times n})$.

1 Introduction

In many applications, one has to find eigenvalues of symmetric positive definite matrices depending on parameters, and/or singular values of general matrices. For example, this problem arises in the study of equilibria of dynamical systems, in the study of waves propagating on lattices, in the study of adiabatic invariants; e.g., see [4,9,11,13]. In this work, we will be concerned with the Singular Value Decomposition, SVD, of general (non-symmetric) matrix valued functions. Following the approach we developed in [5], the same considerations –and algorithms– we study here apply also to the case of the Schur decomposition of symmetric matrices.

Recall that, for a matrix $A \in \mathbb{R}^{n\times n}$, the SVD is the decomposition $A = U\Sigma V^T$, where $U, V \in \mathbb{R}^{n\times n}$ are orthogonal and Σ is diagonal, with diagonal given by the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$. As it is well known, the SVD reveals a lot of information associated to A: Orthonormal bases for the range and the null space of A, as well as for the range and null space of A^T, can all be obtained easily from the factors of the SVD; see [6].

Example 1.1 An interesting application of the SVD is that it is easy to obtain the best approximation (in the 2-norm) to the matrix A amongst matrices of a given rank. Namely, if $A = U\Sigma V^T$, and $\sigma_1 \geq \cdots \geq \sigma_q > \sigma_{q+1} \geq \cdots \geq \sigma_n$, then the best approximation of rank q to A is given by $U_{:,1:q}\Sigma_{1:q,1:q}\Big(V_{:,1:q}\Big)^T$, that is $\sum_{i=1}^q \sigma_i U_{:,i}(V_{:,i})^T$. Of course, if all singular values are distinct, the SVD at once allows to obtain the complete set of best approximations of rank $q = 1, 2, \ldots, n-1$.

When the matrix A changes with respect to parameters, it is reasonable to assume that A changes smoothly with respect to these parameters. So, mathematically, we will think to have a l–times differentiable function of p parameters, taking values in the set of $(n \times n)$ real matrices: $A \in \mathcal{C}^l(\mathbb{R}^p, \mathbb{R}^{n\times n})$, and usually $l \geq 1$. It is natural to explore, and possibly exploit computationally, the differentiability of the factors in the SVD of A. Unfortunately, except for the case of $p = 1$, the factors in the SVD of A in general do not vary smoothly when singular values of A coalesce. See Section 2 for a brisk overview, and [5] for more details and references.

Moreover, when singular values of the function A, say the k-th and $(k+1)$-st, coalesce, there is an immediate lack of uniqueness in the matrix of best approximation of rank k to A. In this sense, coalescing singular values represent (local) bifurcation phenomena. And, similarly to what one does when studying bifurcation phenomena, it is thus important to locate parameter values where singular values coalesce: *coalescing points*. This is our scope in this paper: To accurately locate coalescing points.

We stress that accurately locating parameter values where singular values coalesce is more than an interesting and challenging problem. In the words of Yarkony (see [13], apropos eigenvalues of symmetric matrices), "... what was once viewed largely as a theoretical curiosity is an essential aspect of electronically nonadiabatic processes", and indeed the extensive bibliography in [13] dedicated to this phenomenon (eigenvalues coalescing) is a testament to its practical impact. So much so, that techniques have been devised to attempt locating coalescing points; again, see [13] and the bibliography therein.

Our algorithmic approach is not only new and more general than those reported in the above work, but it presents some distinct advantages. First, it is better capable to locate general coalescing points, not just conical intersections. Second, our "localization" phase of the algorithm is not based on perturbative arguments, and is thus global in nature; indeed, it is similar in spirit to the bisection method to locate a root of a single nonlinear equation. Finally, thanks to the "zoom-in" module of our numerical method, our algorithm is locally quadratically convergent; cfr. the results in [13], which betray linear convergence rate. The price we pay for these apparent enhancements is that we need a reliable solver for the SVD of a 1-parameter function. This 1-d solver must return approximations of the factors of the underlying smooth factorization; this is not trivial, but it has been examined in other works and we will borrow an approach implemented by the authors of [4].

A plan of the paper is as follows. In Section 2, we provide necessary theoretical background, and discuss genericity issues. The main results are Theorems 2.6 and 2.7 which form the basis for our algorithms. The algorithms are discussed in Section 3 and the localization and zoom-in phases are explained in some detail. The performance of our techniques is illustrated in the Examples of Section 4. Conclusions are in Section 5.

Remark 1.2 The SVD is well defined also for rectangular matrices $A \in \mathbb{R}^{m\times n}$, with $m \geq n$. Indeed, in this case $A = U\Sigma V^T$, where $U \in \mathbb{R}^{m\times n}$ is orthonormal ($U^TU = I$), $V \in \mathbb{R}^{n\times n}$ is orthogonal, and $\Sigma \in \mathbb{R}^{n\times n}$ is diagonal with the ordered singular values along the diagonal. We can easily consider this case, if we restrict to full rank functions A; that is, the function $A \in \mathcal{C}^l(\mathbb{R}^p, \mathbb{R}^{m\times n})$ is full rank. In fact, in this case, the Gram-Schmidt procedure of A produces $A = QR$ with $Q \in \mathcal{C}^l(\mathbb{R}^p, \mathbb{R}^{m\times n})$ orthonormal, and $R \in \mathcal{C}^l(\mathbb{R}^p, \mathbb{R}^{n\times n})$ triangular, and we are left to work with a square function (R, here). For this reason, we just restrict to square valued functions in this work.

2 Background

A very important issue to address, when dealing with functions at large, is that of genericity. Loosely speaking, a generic property is a property that one expects to hold for a "general function", and that persists under small perturbations (see [8] for background on this concept). Addressing this issue helps distinguishing between what is a pathological behavior from what is a "regular" behavior and should be expected. Besides this, considering generic properties is of primary importance from a numerical point of view, as it is vain to attempt to numerically detect a phenomenon which may even disappear under arbitrarily small perturbations.

In 1929, von Neumann and Wigner gave a fundamental result (see [12]) which is directly related to the genericity of having a pair of coalescing singular values. Their result, known as *non-crossing rule*, states that having a pair of coalescing eigenvalues for a real symmetric matrix valued function, and hence a pair of coalescing singular values for a general matrix valued function, is a codimension-2 phenomenon. For our purposes, the main consequence of this fact is that having a pair of coalescing singular values is a generic property for matrix valued functions depending on two (or more) parameters, but not for functions depending on 1 parameter. More precisely, we should not expect a matrix valued function depending on 1 parameter to have coalescing singular values, and in case a pair of singular values do coalesce, the coalescing would break down under an arbitrarily small perturbation. On the other hand, for 2-parameter matrix functions, the result says that coalescing of singular values must be expected, that it should occur at isolated points in parameter space, and that it should persist under small perturbations.

2.1 Breaking Coalescing Singular Values Under Perturbation

Studying the way degenerate cases break down under perturbation turns out to be quite insightful, and understanding this situation is also of help in understanding how the algorithms work. For these reasons, we look at this situation here below. We restrict to (2×2) cases, so to keep the arguments simple without losing any essential feature. First, consider the 1-parameter case.

Example 2.1 Let $t \in [0, 1]$,

$$\Sigma(t) = \begin{bmatrix} 1 + (t - .5)^2 & 0 \\ 0 & 1.125 \end{bmatrix}, \quad U(t) = \begin{bmatrix} \cos(t) & \sin(t) \\ \sin(t) & -\cos(t) \end{bmatrix}$$

and $A(t) = U(t)\,\Sigma(t)\,U^T(t)$. The singular values of A coalesce for two values of the parameter t, see Figure 1. Note that, in order to preserve the smoothness of the SVD factors, singular values have been allowed to exchange ordering. Let us perturb A by

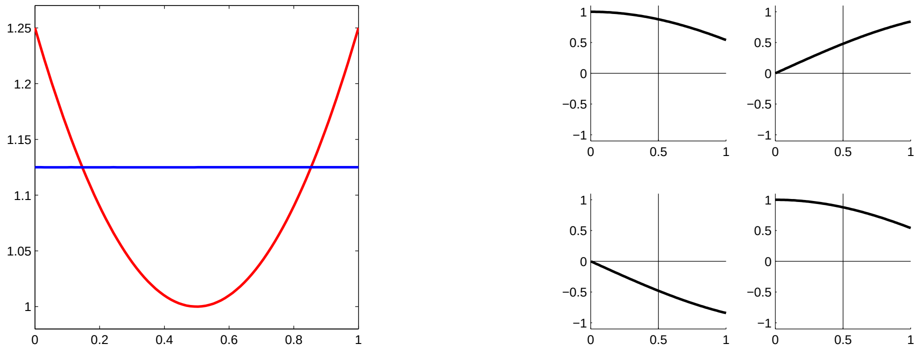

Fig. 1. Singular values of A (left) and each of the entries of one matrix of singular vectors (right).

adding to it a (2×2) matrix E, with entries chosen at random (uniformly distributed in $[-1, 1]$) and of small norm:

$$E = 10^{-2} \begin{bmatrix} 0.843... & -0.647... \\ 0.476... & -0.188... \end{bmatrix}.$$

Figure 2 shows the effect of the perturbation on the singular values and singular vectors. As we expected, the singular values of $A + E$ do not coalesce. Actually, they come very close to coalescing, but then they suddenly “curve away”. This phenomenon has been often observed in applications (see [9,11]), and is known as “veering” or “avoided crossing”. We remark that this phenomenon is not an anomaly, but rather a typical behavior for 1-parameter dependent matrix functions. Something very interesting occurs concerning the singular vectors: Their entries seem to develop nearly jump discontinuities in correspondence of the points where the singular values of the unperturbed matrix A coalesce. This is also what one should expect to observe in general. But, while the “veering” effect is easily justified by a codimension argument, the “jump” behavior of the singular vectors deserves more explanations. In Example 2.3 below we show that it is a side-effect of the breakdown of a coalescing pair of singular values, essentially due to a sharp 90° rotation of the corresponding singular vectors.

Remark 2.2 The results whose outcome we reported in Figures 1 and 2 have been obtained with the 1-d solver which computes a smooth path of SVD decompositions; see Section 3.1.1. It is noteworthy to point out that the computations for the perturbed case are far more expensive and delicate than those for the unperturbed case (at first, this appears to be almost counterintuitive!). The reason is that the closer one gets to the coalescing point of the unperturbed problem, the harder one has to work to maintain smooth decompositions for the ortogonal factors. This is due to the high speed of rotation of these factors near the coalescing point of the unperturbed problem. To witness, to complete the smooth path of SVDs for the two cases of Figures 1 and 2 required 85, respectively 670, continuation steps.

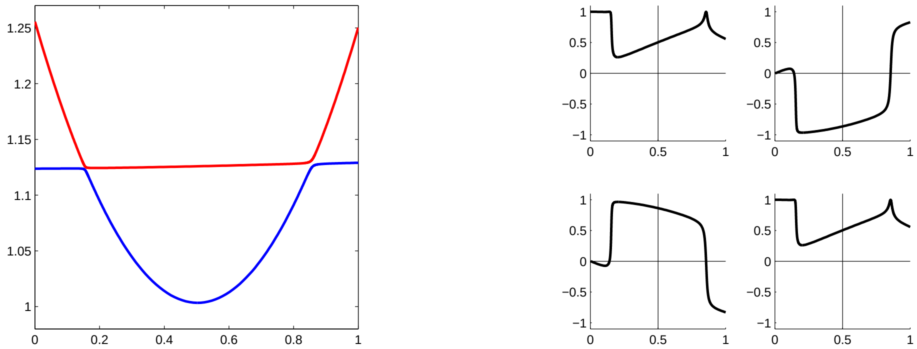

Fig. 2. Singular values of $A + E$ (left) and each of the entries of one matrix of singular vectors (right).

Example 2.3 (Jump of Singular Vectors) Let $A \in \mathcal{C}^l(\mathbb{R}, \mathbb{R}^{2\times 2})$, and let its singular values σ_1, σ_2, and corresponding singular vectors u_1, u_2, v_1, v_2, be all smooth functions. Suppose $\sigma_1(t) = \sigma_2(t)$ if and only if $t = \bar{t}$. Let $E \in \mathcal{C}^l(\mathbb{R}, \mathbb{R}^{2\times 2})$ be such that $E(t) = 0$ for all $|t - \bar{t}| > \delta$, for some small $\delta > 0$. We will think of E as a (compactly supported) perturbation function. Let $\widetilde{\sigma}_1$ and $\widetilde{\sigma}_2$ be the smooth singular values of the "perturbed" function $A + E$ and $\widetilde{u}_1$, $\widetilde{u}_2, \widetilde{v}_1$, $\widetilde{v}_2$, the corresponding smooth singular vectors. Now assume that $\widetilde{\sigma}_1(t) \neq \widetilde{\sigma}_2(t)$ for all t (in particular, near $t = \bar{t}$). In other words, we are assuming that the perturbation actually breaks down the coalescing. This last assumption is not at all restrictive, since we expect that just about every E will achieve it. The argument that follows is based on simple observations on the degree of uniqueness of the SVD of a matrix with distinct singular values. Since $A(t) = A(t) + E(t)$ for all $t < \bar{t} - \delta$, for the same values of t we have that $\sigma_1(t) = \widetilde{\sigma}_1(t)$ and $\sigma_2(t) = \widetilde{\sigma}_2(t)$. The degree of non-uniqueness of the corresponding singular vectors is given by their signs. Therefore we can, and do, assume that $u_1(t) = \widetilde{u}_1(t)$, $u_2(t) = \widetilde{u}_2(t)$, for $t < \bar{t} - \delta$, and the same thing for the right singular vectors (v_1, v_2 and $\widetilde{v}_1, \widetilde{v}_2$). For $t = \bar{t}$, $\sigma_1(t)$ and $\sigma_2(t)$ coalesce and cross each other, while $\widetilde{\sigma}_1(t)$ and $\widetilde{\sigma}_2(t)$ do not coalesce for any value of t, see Figure 3. On the other hand, we have $A(t) = A(t) + E(t)$ for all $t > \bar{t} + \delta$. Hence, for the same values of t, we must have $\sigma_1(t) = \widetilde{\sigma}_2(t)$, $\sigma_2(t) = \widetilde{\sigma}_1(t)$, $u_1(t) = \pm\widetilde{u}_2(t)$, $u_2(t) = \pm\widetilde{u}_1(t)$, for $t > \bar{t} + \delta$, and the same thing for the right singular vectors. In other words, the singular vector $\widetilde{u}_1$ (resp. $\widetilde{u}_2$), which starts off aligned to u_1 (resp. u_2), rotates toward the direction of u_2 (resp. u_1) over the interval $[\bar{t} - \delta, \bar{t} + \delta]$. Since $\widetilde{u}_1\ \widetilde{u}_2$ is a smooth orthonormal frame for all $t \in \mathbb{R}$, the only (two) possibilities, for $t > \bar{t} + \delta$, are:

$$\begin{bmatrix} \widetilde{u}_1 & \widetilde{u}_2 \end{bmatrix} = \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} 0 & \pm 1 \\ \mp 1 & 0 \end{bmatrix}.$$

The same thing can be argued concerning the right singular vectors. This justifies our claim: If a 1-parameter matrix function is perturbed so that a coalescing of singular values breaks down, the corresponding singular vectors will perform a sharp ± 90° rotation. Note that the rotation occurs over an interval of length 2δ, where $\delta > 0$ is arbitrarily small. Therefore, in principle, the rotation may be made as sharp as

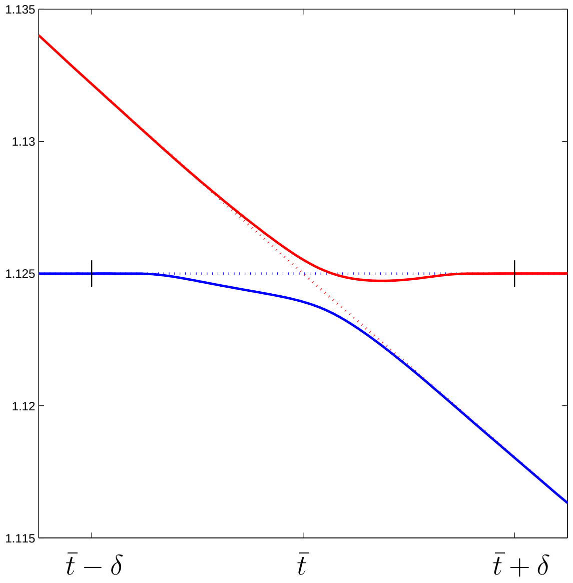

Fig. 3. Dotted lines represent σ_1 (red) and σ_2 (blue), solid lines represent $\widetilde{\sigma}_1$ (red) and $\widetilde{\sigma}_2$ (blue); the Figure also shows the support of the perturbation

desired.

Remark 2.4 In applications, one often knows the value of the function $A(\cdot)$ only at a discrete set of points $t_0, \dots, t_N$, and it is natural to try to find an approximation to the function A via cubic splines or some other interpolation tool. More in general, it is easy to imagine situations where A is known only up to a certain error. Example 2.1 clearly shows that it would be vain to attempt locating coalescing points in such cases.

In the next example we illustrate how a degenerate situation for a two parameter function breaks down under perturbation.

Example 2.5 Let $x_1, x_2 \in [-1, 1]$,

$$A(x_1, x_2) = \begin{bmatrix} x_1^2 + x_2^2 & 0 \\ 0 & .64 \end{bmatrix} .$$

The two singular values of A are given by a paraboloid-shaped surface and a horizontal flat plane, which coalesce along the circle of radius .8 centered at the origin, see Figure 4. This is clearly a degenerate situation since, by the result of von Neumann and Wigner, singular values of a 2-parameter dependent matrix function are expected to coalesce at isolated points. Just as in Example 2.1, let us perturb the matrix function A by adding a small perturbation matrix E to it. As Figure 5 shows, after the perturbation the singular values of $A + E$ do actually coalesce only at four isolated points (three of which are visible in the figure). If we zoom-in on one of the coalescing points (see Figure 6), we can better understand the nature of the contact: The two surfaces of singular values intersect in a double-cone fashion. This turns out to be a typical behavior (see [5], [12], [13]) for coalescing singular values of matrices

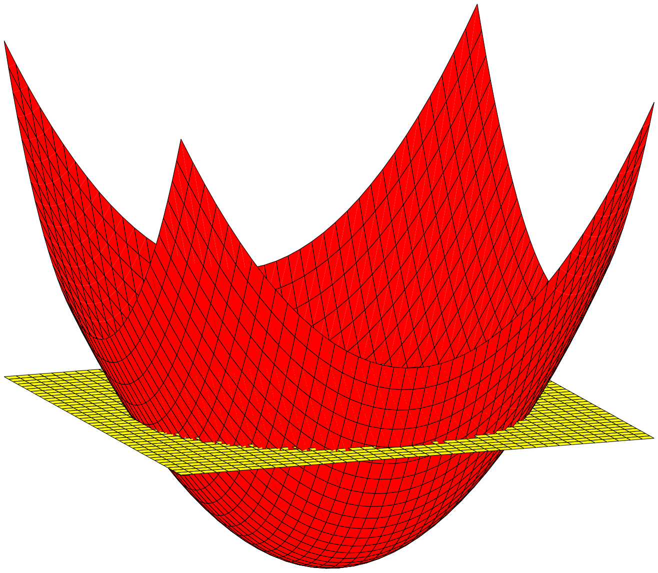

Fig. 4. Singular values of the "unperturbed" matrix A.

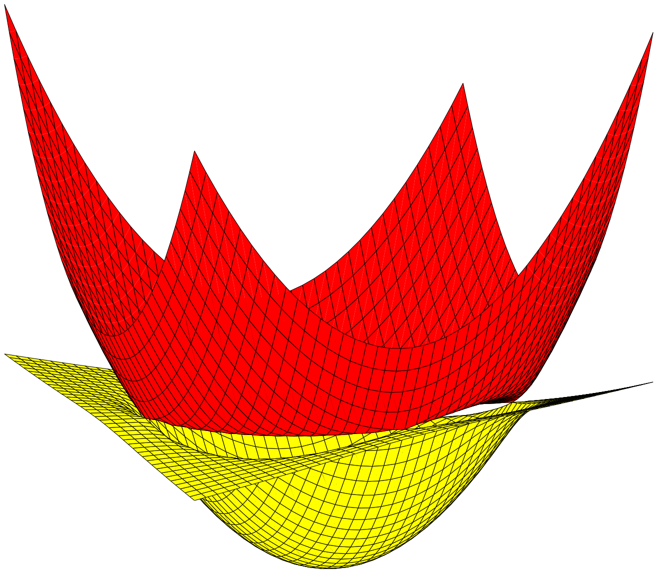

Fig. 5. Singular values of the "perturbed" matrix $A + E$.

depending on two parameters, and justifies the name of *conical intersection* [1] that is given to such phenomena.

[1] See the nice description at http://en.wikipedia.org/wiki/conical_intersection

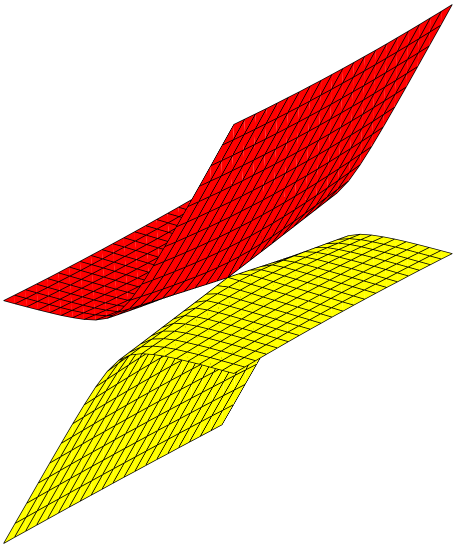

Fig. 6. Zoom-in on a "conical intersection".

2.2 Theoretical Results

We finish this section by giving two results which form the backbone of our algorithm for the localization of coalescing points for 2-parameter dependent matrix functions. Both results are adapted from [5], to which we refer for proofs and further details.

Theorem 2.6 *Let $A \in \mathcal{C}^l(\Omega, \mathbb{R}^{n\times n})$, $l \geq 1$, $\Omega \subseteq \mathbb{R}^2$ simply connected. For all $x \in \Omega$, let $\sigma_1(x) \geq \ldots \geq \sigma_n(x)$ be the continuous singular values of $A(x)$, and suppose that, for every $k = 1, \ldots, n-1$,*

$$\sigma_k(x) = \sigma_{k+1}(x)$$

at $d_k \geq 0$ generic[2] distinct points in Ω. Let Γ be a simple closed curve enclosing (in its interior) all these coalescing points, parametrized by the 1-periodic function $\gamma \in \mathcal{C}_1^p(\mathbb{R}, \Omega)$. Let $m = \min(l, p)$ and $A_\gamma \in \mathcal{C}_1^m(\mathbb{R}, \mathbb{R}^{n\times n})$ defined as $A_\gamma(t) = A(\gamma(t))$, for all $t \in \mathbb{R}$. Then, for all $t \in \mathbb{R}$,

$$A_\gamma(t) = U_\gamma(t) S_\gamma(t) V_\gamma^T(t) \ ,$$

where:

(i) $S_\gamma \in \mathcal{C}_1^m(\mathbb{R}, \mathbb{R}^{n\times n})$ diagonal,
(ii) $U_\gamma, V_\gamma \in \mathcal{C}^m(\mathbb{R}, \mathbb{R}^{n\times n})$ orthogonal, with

$$U_\gamma(t+1) = U_\gamma(t)\ D \ , \ V_\gamma(t+1) = V_\gamma(t)\ D \ , \quad \forall t \in \mathbb{R} \ ,$$

[2] See Definition 3.2 in [5] for the definition of generic points in this contest

and D is a diagonal matrix given by:

$$D_{11} = (-1)^{d_1}, \quad D_{kk} = (-1)^{d_{k-1}+d_k} \text{ for } k = 2, \dots, n-1, \quad D_{nn} = (-1)^{d_{n-1}} \ .$$

The next Theorem is the converse of Theorem 2.6 just stated.

Theorem 2.7 *Let A, Γ, γ and D be just as in Theorem 2.6. Let $2q$ be the (even) number of indices k_i for which $D_{k_ik_i} = -1$. Let us group these indices in pairs $(k_1, k_2), \dots, (k_{2q-1}, k_{2q})$. Then, generically, σ_k and σ_{k+1} coalesce at an odd number of points counting multiplicities*[3] *inside the region encircled by Γ, if and only if $k_{2j-1} \le k < k_{2j}$ for some $j = 1, \dots, q$.*

Basically, the two theorems establish a correspondence between coalescing of singular values and periodicity of the SVD-factors as one completes a loop along Γ.

Remark 2.8 We illustrate how Theorem 2.7 could, and will, be used as a tool in detecting regions in which singular values coalesce. Let $A \in \mathcal{C}^l(\Omega, \mathbb{R}^{n\times n})$, $l \ge 1$, $\Omega \subset \mathbb{R}^2$ simply connected. Suppose we have computed the continuous SVD of A along a closed simple curve $\Gamma \subset \Omega$, without having encountered any singular values coalescing along the path. Suppose that, after a complete loop along Γ, the singular vectors associated to σ_k and σ_{k+1} have changed sign (getting "upside down") while all other singular vectors have come back to their original position. Then, Theorem 2.7 says that σ_k and σ_{k+1} have coalesced at an odd number of points inside Γ. Note that nothing can be inferred about the other pairs, as an even number of coalescing points between the same pair of singular values will go unnoticed (see Example 4.1).

3 Algorithms

We are ready to describe our algorithms. We will restrict consideration to the case when Ω is a rectangle: $\Omega = \{x = (x_1, x_2) \in \mathbb{R}^2 \ : \ a \le x_1 \le b \ , \ c \le x_2 \le d \ \}$. The case of a triangular region is handled in a similar way, and the case in which Ω is a simply connected region with piecewise linear boundary is also handled with no major modifications. The case in which Ω is a disk is also simple, and one of our examples at the end considers this case. Through appropriate subdivisions, and reparametrizations, we can, in principle, tackle more complicated situations.

Our algorithm consists of two main modules: localization and zoom-in. Throughout, we assume that coalescing of singular values occurs only at isolated points.

[3] See Definition 3.9 in [5] for the definition of multiplicity in this contest

The localization module attempts to locate subregions of Ω where there is a *coalescing point*. By this, we mean a conical intersection, or more generally an intersection of odd multiplicity, of singular values. The technique we use is based on the theoretical results given in Section 2, see Theorem 2.7, which can be rephrased as follows: The presence of a coalescing point inside a region bounded by a closed curve triggers period doubling of the orthogonal factors of the continuous SVD of A computed along the curve. This period doubling phenomenon is what we will use as evidence for the existence of a coalescing point inside a certain region. Let us describe the localization module of the algorithm in more detail.

First, we subdivide Ω in a cartesian grid of $N \times M$ boxes, not necessarily all of the same size. That is, we let $\Omega = \cup_{i=1:N,j=1:M} B_{ij}$, where $B_{ij} = \{(x_1, x_2) \ : \ x_1^{(i)} \leq x_1 \leq x_1^{(i+1)} \ , \ x_2^{(j)} \leq x_2 \leq x_2^{(j+1)} \}$, $i = 1, \dots, N$, $j = 1, \dots, M$. Let us call $x_{i,j} = (x_1^{(i)}, x_2^{(j)})$, and similarly $x_{i+1,j}$, $x_{i,j+1}$ and $x_{i+1,j+1}$, the vertices of the box B_{ij}.

Remark 3.1 Since coalescing points are isolated, we may as well assume that no coalescing point is on the grid itself. However, this eventuality (albeit not to be expected) would actually be easily detected; see Remark 3.2 below.

The idea is to "sweep" the grid, searching for boxes inside which there is a coalescing point. To decide upon the presence in a box of a coalescing point, we monitor whether or not the following condition is satisfied: Going around the box once, the orthogonal factors of a continuous SVD at the beginning of the loop and those at the end of the loop do not match. Here is how we do it.

Let us consider the general box B_{ij}. We think of the boundary of B_{ij} as made up of two L-shaped paths, Γ_1 joining $x_{i,j}, x_{i+1,j}, x_{i+1,j+1}$ and Γ_2 joining $x_{i,j}, x_{i,j+1}, x_{i+1,j+1}$; see Figure 7.

Note that both paths have the same starting and end points. We compute the continuous SVD of A along both paths, starting from the same reference decomposition $A(x_{i,j}) = U_0 \Sigma_0 V_0^T$, and record the orthogonal factors U at the end of each path, call them $U_1(x_{i+1,j+1})$ and $U_2(x_{i+1,j+1})$. At this point, the two orthogonal factors are compared. If they match, nothing can be inferred about the existence of a coalescing point inside the box, and we either refine the box further or –if the size of the box is deemed sufficiently small– we declare that there is no coalescing point within the desired resolution. On the other hand, if they do not match, we look at the matrix $D := \text{diag}\Big(U_1(x_{i+1,j+1})^T \ U_2(x_{i+1,j+1})\Big)$, which contains the information about what pair of singular values have coalesced inside the box under consideration; this is the same D as in Theorems 2.6 and 2.7.

This procedure is followed until all boxes have been explored. Information about which pairs of singular values have coalesced in which boxes is saved and will serve

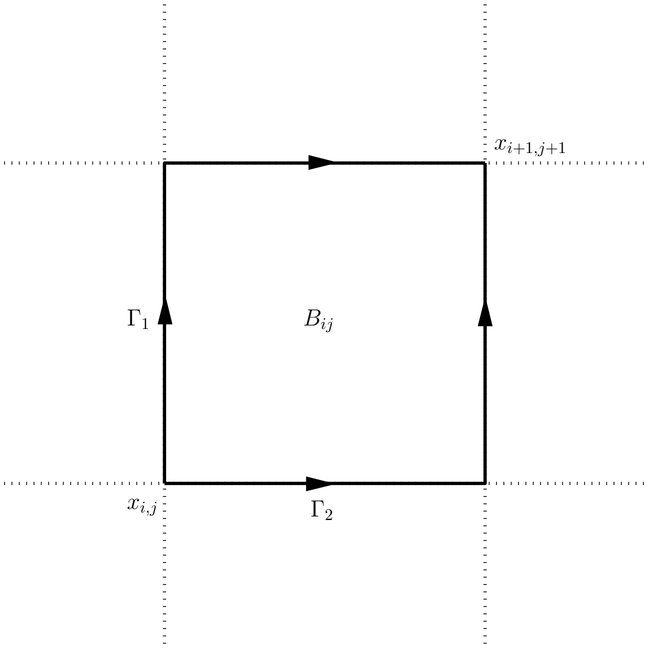

Fig. 7. The paths Γ_1 and Γ_2 in a typical box.

as input for the zoom-in module.

"Refinement". Obviously, the localization procedure above can be implemented so to refine a box where there is a coalescing point. For example, progressively halving each vertical/horizontal side ("bisection" approach). By doing so, in principle, one would be able to locate a coalescing point with arbitrary degree of accuracy. Still, it is pretty obvious that this refinement process will converge only linearly, and this is one of the reasons why we prefer a quicker zoom-in technique once we have a good initial guess.

3.1.1 1-d Solver

A separate component of our techniques rests on the ability to compute the continuous SVD along the sides of a box, say the paths $\Gamma_{1,2}$ of which above. We describe the procedure relatively to computation of the path of SVDs along Γ_1 (obviously, the same applies to Γ_2). First of all, we force the path to step exactly (from x_{ij}) at both $x_{i+1,j}$ and to $x_{i+1,j+1}$, so that it is enough to describe the procedure on the line segments between these vertices, say between x_{ij} and $x_{i+1,j}$. We can thus think that we need to compute a smooth path of SVDs of the $\mathcal{C}^l([0,1],\mathbb{R}^{n\times n})$ function A, and we want $\mathcal{C}^l$ factors U, Σ, V. (The normalization to $[0,1]$ is for convenience only, obviously the physical interval in the case under consideration is the difference of the first components of $x_{i+1,j}$ and x_{ij}.)

It is absolutely crucial that the SVD be continued along each path in a continuous way, as otherwise we cannot make reliable inferences at the end point $x_{i+1,j+1}$. Assuming distinct singular values, it is easy to produce smoothly varying factors along the path of SVDs. Consider the following.

Let $0 = t_0 < t_1 < \cdots < t_N = 1$ be a partition of $[0, 1]$, and let $h_j = t_{j+1} - t_j$, $j = 0, 1, \ldots, N-1$. In practice, the stepsizes h_j, $j = 1, \ldots, N-1$, will be found adaptively, see below. The first stepsize h_0 is arbitrary and our code takes $h_0 = 10^{-3}$, but the adaptive stepsize strategy quickly adjusts it. We have $A(0) = U_0\Sigma_0V_0^T$ as our starting decomposition at t_0.

The basic technique consists of the following steps, which have been implemented by the second and third authors of [4] [4] .

1. Using canned software, we compute the SVD at the points t_j, $j = 1, 2, \ldots, N$, that is $A(t_j) = U_j\Sigma_jV_j^T$. Since the singular values of $A(t_j)$ are distinct, and ordered along the diagonal of Σ, then the orthogonal factors U_j and V_j are uniquely determined except for joint changes of signs of the columns of U_j and V_j. In order to guarantee continuity of the factors, we choose the signs of these columns in agreement with the signs of the columns of the factors in the *predicted* values at t_j, call these $U_j^{(0)}, V_j^{(0)}$. The latter values, the predicted values, are obtained by extrapolating at t_j the line through (t_{j-2}, U_{j-2}) and (t_{j-1}, U_{j-1}), and similarly for V. Then, we adjust the signs of U_j so that $(U_je_k)^T(U_j^{(0)}e_k) > 0$, for all $k = 1, \ldots, n$. For the first step, that is at t_1, the predicted values are just the old values at t_0.

2. To choose the stepsizes adaptively, we use the same strategy of [4]. For simplicity, here below we omit the index j indicating the gridpoint t_j at which we are. We have predicted values $U^{(0)}, \Sigma^{(0)}, V^{(0)}$, and *corrected* values U, Σ, V, so that we can compute the errors $\|\Sigma^{(0)} - \Sigma\|$, $\|U^{(0)} - U\|$ and $\|V^{(0)} - V\|$. Next, we compute the weighted norm $\rho_\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{\sigma_i^{(0)}-\sigma_i}{\epsilon_r|\sigma_i|+\epsilon_a})^2}$, where ϵ_r and ϵ_a are relative and absolute error tolerances, and analogously we compute ρ_U, ρ_V. [In our experiments, we use $\epsilon_r = \epsilon_a = 100\ \texttt{EPS}$, where $\texttt{EPS} \approx 10^{-16}$ is the machine precision.] Then, we set $\rho = \max(\rho_\sigma, \rho_U, \rho_V)$ and $h_{\text{new}} = \frac{h}{\sqrt{\rho}}$. If $\rho \leq 1.5$, the step is accepted, otherwise it is rejected. In all cases, h_{new} is used as the new steplength. Finally, the last stepsize is chosen so to land exactly at $t_N = 1$.

Remark 3.2 If there are coalescing points, they can be easily detected by the 1d-solver, by looking at the ordering of the singular values in the predicted $\Sigma_j^{(0)}$: If these are not ordered, then we can infer which pair must have exchanged ordering and –as in [4]– we can trigger a one-dimensional secant search to locate the coalescing point precisely.

[4] We thank M.G. Gasparo and A.Papini for letting us use a `Matlab` version of their program

3.1.2 Implementation

We begin with a cartesian grid, which defines boxes B_{ij}, $i = 1, \dots, N$, $j = 1, \dots, M$. Our basic algorithm consists of going through the grid by covering one box at the time. In our implementation, we do this horizontally Left-to-Right and vertically Bottom-to-Top, across the grid. We transverse each edge of any box only once, and never compute twice the SVD at any given point.

As soon as all boxes are covered, and assuming that a coalescing value has been located, we either refine the box or trigger the zoom-in module. We found it effective to refine until the box is sufficiently small that Newton's method (on which the zoom-in module is based) can be estimated to converge quickly (say, in 4 to 5 iterations); see Example 4.2. Assuming that the boxes are squares, a good rule of thumb we have adopted would be to refine until the center of the box has distance from the vertices between 1/10 and 1/100. An important aspect to consider is that there is a trade-off between refining so to get closer to the coalescing point (and then zoom-in on it quickly) and expense. In fact, recall Remark 2.2, the computational work increases considerably near coalescing points, and so progressive refinements become increasingly more expensive because of the fast variation of the orthogonal factors near a coalescing point.

As already remarked, the workhorse of our algorithms is the adaptive continuation code which computes the smooth SVD along the edges of the boxes. This code is simple and fairly robust, but there is certainly room for improvement in the context of interest to us. For example, at present, the code visits the boxes one at the time, which forces frequent restarting and it is an unpleasant overhead for the adaptive step-size selection. Also, the basic algorithm we use is inherently very parallelizable, but at present we have not exploited this feature.

3.2 Zoom In

Given a box B_{ij} in Ω and a pair of consecutive singular values which are known to coalesce in B_{ij}, we want to accurately locate the values of the parameter where the singular values coalesce.

Assume a coalescing between σ_k and σ_{k+1} has been detected in the box B_{ij}. Then we apply Newton's method to minimize the nonlinear functional

$$f(x) = (\sigma_k(x) - \sigma_{k+1}(x))^2 .$$

To be precise, since the function f is a smooth paraboloid-shaped function, we look for a solution of the problem $\nabla f(x) = 0$, rewritten as $F(x) = 0$, where $F : \mathbb{R}^2 \to \mathbb{R}^2$. On this systems, we perfom Newton's method, using the center of the box as the initial guess. Obviously, the method will converge with quadratic rate if the initial guess is close enough to the exact solution.

3.2.1 Implementation

We summarize the algorithm below. As `INPUT`, we give a convergence tolerance `TOL` (we use `TOL` $= 100$ `EPS`), and a maximum number of iterations `MAXIT` (we use 5), as well as the box B_{ij} under scrutiny and the pair of singular values which are known to coalesce in the box; say, $(k, k+1)$.

```
J=0, $x = \mathrm{center}(B_{ij})$.
While $\|F(x)\| >$ TOL, and J<MAXIT, do
$DF(x)y \;=\; -F(x)$          [Solve for $y$]
$x = y + x$, J=J+1, enddo
```

Remarks 3.3 Some observations are in order.

(i) First of all, since the singular values themselves are not expected to be differentiable, while $f(x)$ is, we form $F(x)$ and $DF(x)$ by differencing on f, as follows. For the gradient, we use 2-nd order divided differences

$$F(x) \approx \begin{bmatrix} \Big(f(x_1+h,x_2) - f(x_1-h,x_2)\Big)/(2h) \\ \Big(f(x_1,x_2+h) - f(x_1,x_2-h)\Big)/(2h) \end{bmatrix} .$$

For the second derivatives making up $DF(x)$ we also use centered differences approximations. E.g.,

$$f_{x_1,x_1} \approx \Big(f(x_1+h,x_2) - 2f(x_1,x_2) + f(x_1-h,x_2)\Big)/h^2 \ ,$$

and

$$f_{x_1,x_2} \approx \frac{f(x_1+h,x_2+h) - f(x_1+h,x_2-h)}{4h^2} - \frac{f(x_1-h,x_2+h) + f(x_1-h,x_2-h)}{4h^2} .$$

Therefore, any Newton step requires (and costs as much as) nine SVDs. At present, we have not implemented any strategy to reduce this cost, since we have been chiefly interested in studying feasibility and reliability of the technique on small dimensional problems; of course, for large dimensional problems, one will need to implement new techniques whereby only the relevant singular values' information is computed.

(ii) Newton's method occasionally converges to a solution outside of the box B_{ij}, or fails to converge. This usually betrays a poor initial guess and triggers a refinement procedure for the box B_{ij}.

4 Examples

In this section we exemplify performance of the algorithms on several test problems. We exemplify the following aspects: (i) Impact of the initial subdivision; (ii) Impact of refinement on the zoom-in module; (iii) Adaptation to a different geometry (non-rectangular), and (iv) Performance of the algorithms on degenerate cases. All algorithms have been implemented in `Matlab` and run on a Laptop computer with clock speed of 1.5 Ghz.

Example 4.1 Let $\Omega = \{(x_1, x_2) \in \mathbb{R}^2 \ : \ -1 \le x_1 \le 1 \ , \ -1 \le x_2 \le 1 \ \}$,

$$B(x_1, x_2) = \mathrm{diag}(x_1^2 + x_2^2 \ , \ .81 \ , \ .36),$$

$$C = \begin{bmatrix} -0.179... & -0.294... & -0.722... \\ 0.787... & 0.626... & -0.594... \\ -0.884... & -0.980... & -0.602... \end{bmatrix},$$

$$E(x_1, x_2) = .5(x_1 + x_2)(x_1 + 1/3)\, C\,,$$

and $A(x_1, x_2) = B(x_1, x_2) + E(x_1, x_2)$, for all $(x_1, x_2) \in \Omega$. The entries of C have been chosen at random (uniformly distributed in $[-1, 1]$). Just as in Example 2.5, the matrix function A has been built as the sum of a matrix B, whose singular values coalesce in a degenerate way (along two circles centered at the origin of radius .6 and .9) and a perturbation matrix E. In order to force the existence of some coalescing points, the perturbation matrix E has been chosen so that $E = 0$ along two lines in the parameter space, which intersect the above-mentioned circles at eight points. Therefore, at least eight coalescing points are known to exist in Ω (their coordinates may even be easily computed by hand). Here we report on a few experiments using different choices of grids (all uniform). First, we have run the localization module subdividing Ω into a 2×2 grid. Coalescing points for the pair σ_2-σ_3 have been detected in boxes B_{11} and B_{21}, see Figure 8. Switching to a 10×10 grid, coalescing points are detected in 10 boxes, see Figure 9. [Note that 8 coalescing points had gone undetected through the 2×2 grid due to the fact that they occur in pair over each box.] When we run the zoom-in module on each of the ten boxes, we fail to converge to the coalescing points in B_{41}, B_{51}, B_{72} and B_{92}. Finally, we repeated the procedure over a 100×100 grid. No further coalescing points are detected, but now the zoom-in module converges to all points. It takes around 75 seconds of cpu-time to complete the search over the 100×100 grid.

Example 4.2 Let us consider again the function A as in Example 4.1. In that Example, the 10×10 grid was enough to detect all coalescing points but not to zoom-in on all of them. Now, instead of switching to a finer grid over the entire region, we will perform a local refinement, only where needed. In this example, we do a bisection-type approach on box B_{41}. We perform 5 successive bisections of each side of the box, thus getting within ≈ 0.0044 of the coalescing point (see shaded

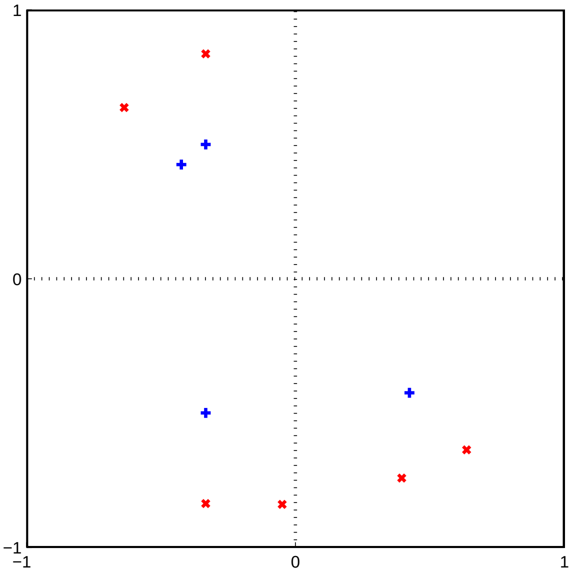

Fig. 8. 2×2 grid; coalescing points are indicated with a $\times$ $(\sigma_1 = \sigma_2)$ or a $+$ $(\sigma_2 = \sigma_3)$

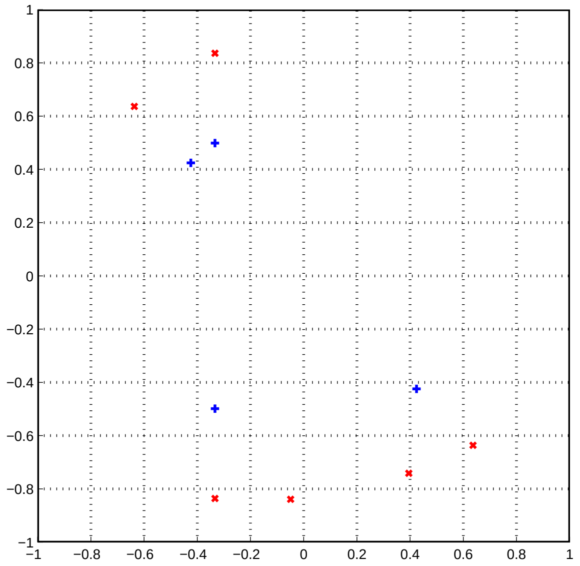

Fig. 9. 10×10 grid; coalescing points are indicated as in Figure 8

region on Figure 10 and beware of the coordinates values). Now we can safely run the zoom-in module and expect to converge to the coalescing point in at most 4 Newton iterates, as indeed happens.

Example 4.3 As already mentioned, our localization technique does not place, in principle, any restriction on the choice of the planar grid. Specific geometric properties of the domain of interest may suggest different choices of grids, other than cartesian. In this example, we consider the same matrix function A as in Examples 4.1 and 4.2, defined on the circular domain $\Omega = \{(x_1, x_2) \in \mathbb{R}^2 \; : \; x_1^2 + x_2^2 \leq 1 \}$.

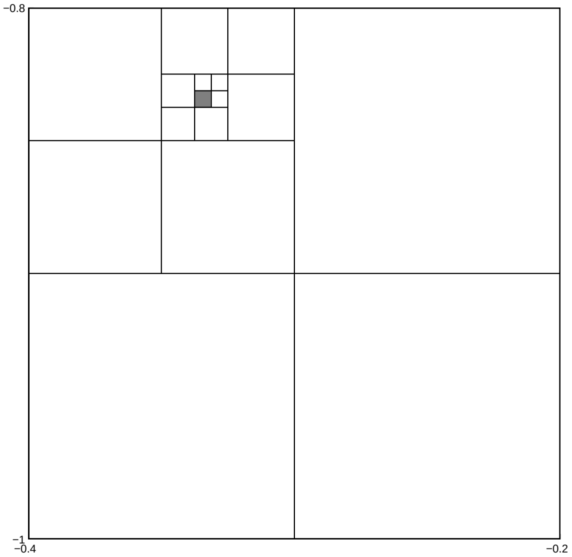

Fig. 10. Local refinement on box B_{41}

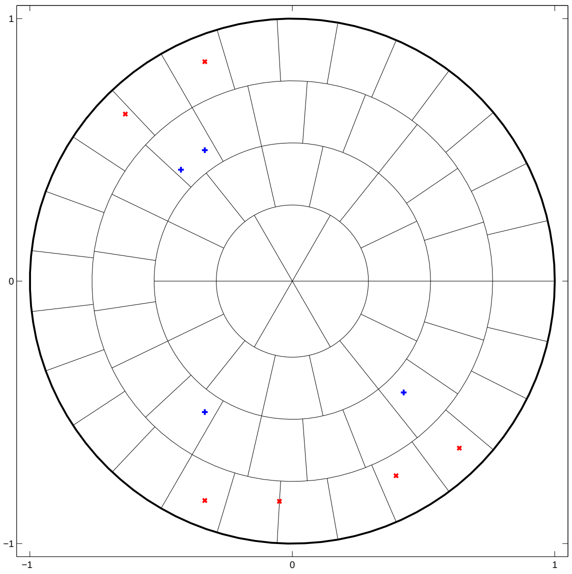

Fig. 11. Non-cartesian grid on a circular domain; coalescing points are indicated as in Figure 8

Given the particular shape of Ω, we decided to use the grid shown in Figure 11. Even though it is not cartesian, the grid can be assimilated to the union of (four) cartesian ones via a change to polar coordinates. Dimension and number of the "tiles" have been chosen so that the distance between the centroid of each tile and any point on it is not larger than 0.2. With this choice of grid, two coalescing points for the pair σ_2-σ_3 go undetected. The localization module locates the remaining eight points, but the zoom-in module fails to converge to two of them.

Example 4.4 This is an example chosen to highlight what happens for degenerate (i.e., non-conical) intersections.

Let $\Omega = \{(x_1, x_2) \in \mathbb{R}^2 \ : \ -1 \leq x_1 \leq 1 \ , \ -1 \leq x_2 \leq 1 \ \}$ and

$$A_p(x_1, x_2) = \begin{bmatrix} x_2 + 2 & x_2 \\ x_2 & x_1^p + 2 \end{bmatrix},$$

where $p \geq 1$ is an integer. For all p, A_p is symmetric positive definite on Ω. Direct computation shows that its eigenvalues coalesce if and only if

$$\begin{cases} x_2 = x_1^p \\ x_2 = 0 \end{cases}, \tag{1}$$

i.e. if and only if $x_1 = x_2 = 0$. Following [5], we define the multiplicity of this coalescing point as the order of contact of the two planar curves defined by the equations (1). Hence, in this example, the origin is a coalescing point of multiplicity p for A_p, for all $p \geq 1$. According to Theorem 2.7, a coalescing point does not produce any period doubling if its multiplicity is even. This causes our localization algorithm to fail to detect coalescing points of even multiplicity. We have tested our localization module on A_p for $p = 1, 2, 3$, and observed what was predicted by the Theorem: The coalescing point is correctly detected for $p = 1, 3$ while goes undetected for $p = 2$. We stress that, in general, one should not expect to have coalescing points of multiplicity $p \geq 2$ for matrix functions depending on two parameters, as the codimension of such phenomena is higher than two.

5 Conclusions

In this work we have proposed, and showed performance of, some new algorithms to locate parameter values where singular values of matrix valued functions of two-parameters coalesce. Our key idea has been to relate the coalescing to the period-doubling of the singular vectors as we complete a loop around the coalescing point. This is in line with our theoretical development of [5] and the present paper is the companion paper of this other work.

Based upon our present study, we believe that our approach is an appropriate way to detect, and accurately locate, parameter values where singular values coalesce. The method is robust, and ultimately fast. However, more implementation effort is neededed to make the technique efficient, especially for large dimensional problems.

References

[1] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. K. Nichols. Numerical computation of an analytic singular value decomposition by a matrix valued function. *Numer. Math.*, 60:1–40, 1991.

[2] J.L. Chern and L. Dieci. Smoothness and periodicity of some matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 22:772–792, 2001.

[3] L. Dieci and T. Eirola. On smooth decomposition of matrices. *SIAM J. Matrix Anal. Appl.*, 20:800–819, 1999.

[4] L. Dieci, M. G. Gasparo, and A. Papini. Continuation of Singular value Decompositions. *Med. J. of Mathematics*, 2:179–203, 2005.

[5] L. Dieci and A. Pugliese. Two-parameter SVD: Coalescing singular values and periodicity. 2007. Submitted. Also www.math.gatech.edu/∼dieci/preps/twoparSVD.pdf.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computations.* The Johns Hopkins University Press, 2nd edition, 1989.

[7] M.W. Hirsch. *Differential topology.* Springer-Verlag, New–York, 1976.

[8] P.F. Hsieh and Y. Sibuya. A global analysis of matrices of functions of several variables. *J. Math. Anal. Appl.*, 14:332–340, 1966.

[9] N.C. Perkins and C.D. Mote Jr. Comments on curve veering in eigenvalue problems, *Journal of Sound and Vibration*, 106:451–463, 1986.

[10] Y. Sibuya. Some global properties of matrices of functions of one variable. *Math. Annal.*, 161:67–77, 1965.

[11] A. Srikantha Phani and J. Woodhouse and N. A. Fleck. Wave propagation in two-dimensional periodic lattices, *J. Acoust. Soc. Am.*, 119:1995–2005, 2006.

[12] J. von Neumann and E. Wigner. Eigenwerte bei adiabatischen prozessen. *Physik Zeitschrift*, 30:467–470, 1929.

[13] David R. Yarkony. Conical intersections: The new conventional wisdom. *J. Phys. Chem. A*, 105:6277–6293, 2001.