SIAM J. Matrix Anal. Appl.
Vol. 34, No. 2, pp. 519–541

APPROXIMATING COALESCING POINTS FOR EIGENVALUES OF HERMITIAN MATRICES OF THREE PARAMETERS[*]

LUCA DIECI[†], ALESSANDRA PAPINI[‡], AND ALESSANDRO PUGLIESE[§]

Abstract. We consider a Hermitian matrix valued function $A(x) \in \mathbb{C}^{n\times n}$, smoothly depending on parameters $x \in \Omega \subset \mathbb{R}^3$, where Ω is an open bounded region of $\mathbb{R}^3$. We develop an algorithm to locate parameter values where the eigenvalues of A coalesce: *conical intersections* of eigenvalues. The crux of the method requires one to monitor the geometric phase matrix of a Schur decomposition of A, as A varies on the surface S bounding Ω. We develop (adaptive) techniques to find the minimum variation decomposition of A along loops covering S and show how this can be used to detect conical intersections. Further, we give implementation details of a parallelization of the technique, as well as details relative to the case of locating conical intersections for a few of A's dominant eigenvalues. Several examples illustrate the effectiveness of our technique.

Key words. coalescing eigenvalues, three-parameter Hermitian functions, Berry phase

AMS subject classifications. 15A18, 15A23, 65F15, 65F99, 65P30

DOI. 10.1137/120898036

1. Introduction.

A Hermitian matrix $A \in \mathbb{C}^{n\times n}$, $A^* = A$, always has a Schur decomposition

$$A \;=\; U\Lambda U^*\,,$$

where $U \in \mathbb{C}^{n\times n}$ is unitary ($U^*U = I_n$), and $\Lambda \in \mathbb{R}^{n\times n}$ is the diagonal matrix of the eigenvalues, which we can assume ordered: $\Lambda = \operatorname{diag}(\lambda_1, \dots, \lambda_n)$, $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_n$. When the eigenvalues are distinct, the eigenvector matrix U is uniquely determined only up to a *phase matrix* $\Phi = \operatorname{diag}(e^{i\phi_j}, j = 1, \dots, n)$, where the *phases* $\phi_j \in \mathbb{R}$, $j = 1, \dots, n$, are arbitrary real numbers. If A has real valued entries and it is symmetric with distinct eigenvalues, then $\Phi = \operatorname{diag}(\pm 1)$.

In general, a Hermitian matrix $A \in \mathbb{C}^{n\times n}$ is expected to have distinct eigenvalues, in that, if not, an arbitrarily small perturbation of it will do. Next, let us consider a Hermitian matrix valued function, for which we still label the eigenvalues so that they are ordered as before, and continuous. In this case, it is well known (e.g., see [8]) that it is a generic property of smooth Hermitian functions of 1 and 2 real parameters to have simple eigenvalues. Indeed, it has been known since the work [16] that having a pair of coalescing eigenvalues for a Hermitian function is a (real) codimension 3 phenomenon. Therefore, we should expect that if a Hermitian function of 3 real parameters has coalescing eigenvalues, they will occur at isolated points—called *conical intersections*—and the eigenvalues would be merely continuous functions (not even once differentiable) at the coalescing point (e.g., we refer to [8] for mathematical

[*] Received by the editors November 7, 2012; accepted for publication (in revised form) by P.-A. Absil February 19, 2013; published electronically May 14, 2013. This work was supported in part under MIUR.

http://www.siam.org/journals/simax/34-2/89803.html

[†] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332 (dieci@math.gatech.edu).

[‡] Department of Industrial Engineering, University of Florence, 50134 Florence, Italy (alessandra.papini@unifi.it).

[§] Department of Mathematics, University of Bari "A. Moro", 70125 Bari, Italy (alessandro.pugliese@uniba.it).

details). This is in sharp contrast to simple eigenvalues (i.e., of multiplicity 1), which retain the full smoothness of the original function A. For completeness, we summarize this fact in the Lemma below, where for a matrix valued function $A : \Omega \to \mathbb{C}^{n\times n}$, continuous with its first k derivatives ($k \geq 0$), we write $A \in \mathcal{C}^k(\Omega, \mathbb{C}^{n\times n})$, and $\Omega \subset \mathbb{R}^3$ indicates an open bounded region of $\mathbb{R}^3$ diffeomorphic to the open unit ball.

LEMMA 1.1 (simple eigenvalues). *Let $A \in \mathcal{C}^k(\Omega, \mathbb{C}^{n\times n})$, $A^* = A$, be given. Let μ_0 be a simple eigenvalue of $A(x_0)$, $x_0 \in \Omega$. Then, there exists a neighborhood V_0 of x_0 and a unique function $\mu(x)$, $x \in V_0$, such that $\mu \in \mathcal{C}^k(V_0, \mathbb{R})$, $\mu(x_0) = \mu_0$, and $\mu(x)$ is an eigenvalue of A for $x \in V_0$.*

Furthermore, let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ be the continuous and ordered eigenvalues of A, and suppose that λ_k (for some $k = 1, \dots, n$) is a simple eigenvalue for all $x \in \Omega$. Then, λ_k is a $\mathcal{C}^k(\Omega, \mathbb{R})$ function.

Proof. Consider the characteristic polynomial of A:

$$p(\lambda, x) = \det(\lambda I - A(x)) \,,$$

for which we know that

$$p(\mu_0, x_0) = 0 \qquad \text{and} \qquad \frac{\partial p}{\partial \lambda}|_{x_0} \neq 0 \,.$$

Then, the implicit function theorem guarantees the existence of a neighborhood V_0 of x_0 and a smooth function $\mu \in \mathcal{C}^k(V_0)$ such that $p(\mu(x), x) = 0$, $x \in V_0$.

As far as the statement about an eigenvalue λ_k which is simple in Ω, we observe that λ_k is surely a continuous function, $\lambda_k : \Omega \to \mathbb{R}$, because it is a root of the characteristic polynomial. Moreover, since it is simple, from the previous reasoning it is a $\mathcal{C}^k$ function in an open neighborhood of any point $x \in \Omega$, hence it is $\mathcal{C}^k$ in Ω. □

Besides being a problem of intrinsic mathematical interest and beauty (e.g., see the classic book by Kato [9]), the phenomenon of coalescing eigenvalues finds applicability in a multitude of different contexts. To name a few applications, the problem is relevant in stability and bifurcation studies (e.g., see [12] for a comprehensive and recent account of local perturbation theory of coalescing eigenvalues), in chemical and quantum physics, where it underpins intersection of potential energy surfaces (see [15, 2, 20]), in structural dynamics studies (e.g., [14]), as well as in numerical studies concerned with best approximation, model reduction, and data compression (e.g., [10, 13]). This makes conical intersections an important singularity to localize and it is for this reason that here we focus on the problem of locating, and accurately approximating, coalescing points for eigenvalues of a smooth Hermitian function A depending on three real parameters: $A \in \mathcal{C}^k(\Omega, \mathbb{C}^{n\times n})$. We will develop and justify a global algorithm suitable for this task. As far as we know, ours is the first algorithmic study for globally localizing and approximating conical intersections.

In [8], we gave theoretical results on coalescing eigenvalues of Hermitian functions, and here we explore the algorithmic implementation of the mathematical results of [8]. There are several important and challenging aspects of the present algorithmic study. Of course, there are geometrical and numerical challenges, dictated by how large the region Ω is, and by the high computational cost of having to repeatedly perform eigendecompositions of A. But, as we will see, there are also other challenges, the chief one of which is related to the delicate task of monitoring (in a "smooth way") the geometric phase matrix of a Schur decomposition of A, as A varies on the surface bounding Ω. In response to these challenges, we will restrict attention to the case of Ω being a cube, and propose techniques which are highly parallelizable through a natural subdivision scheme. Further, we will develop a new 1-dimensional (1-d) solver which approximates the "minimum variation decomposition" (a specific smooth eigendecomposition) of A along a closed path, and renders an accurate approximation to the sought smooth phase matrix at negligible extra cost with respect to that of finding *any* eigendecomposition.

A plan of the paper is as follows. In section 2, we review the main theoretical results which we need, and we further discuss how to perform continuation along a path in parameter space. In particular, we give an interesting result which shows that a predictor-corrector method of nominal order 0 has in fact order 2. In section 3, we discuss the complete algorithm and the subdivision scheme we employed to explore a cubical region Ω. We further discuss implementation details. Finally, in section 4 we show the performance of our technique on several examples.

Notation. By default, we consider the 2-norm for vectors and the induced norm on matrices, though at times we will use also the Frobenius norm $\|\cdot\|_{\mathrm{F}}$ for matrices, $\|A\|_{\mathrm{F}}^2 = \mathrm{trace}(A^*A)$. If $A \in \mathcal{C}^k(\mathbb{R}, \mathbb{C}^{n\times n})$ is periodic of (minimal) period $\tau > 0$, we write it as $A \in \mathcal{C}^k_\tau(\mathbb{R}, \mathbb{C}^{n\times n})$. With $x = (x_1, x_2, x_3)$ we indicate coordinates in Ω. The surface bounding Ω will be denoted with S, and it will therefore be diffeomorphic to the standard 2-sphere. The standard unit vectors are denoted e_1, e_2, e_3.

2. Background and a new 1-d continuation technique.

2.1. Smooth Schur decomposition.

A key component of our algorithm will be the computation of a "smooth" eigendecomposition along a closed curve. We begin by recalling a differential equation model whose solution gives the decomposition we need.

Let Γ be a closed curve in Ω, parametrized by a 1-periodic smooth function $\gamma(\cdot)$. Consider the restriction A_γ of the matrix function A to this curve. So, we have a smooth function $A_\gamma \in \mathcal{C}^k_1(\mathbb{R}, \mathbb{C}^{n\times n})$, $k \geq 1$. Let us further assume that the function A has distinct eigenvalues along Γ. Under these assumptions, given an initial decomposition $A_\gamma(0) = U_0\Lambda_0U_0^*$, we seek a $\mathcal{C}^k$ Schur decomposition of A_γ: $A_\gamma = U\Lambda U^*$, where $U \in \mathcal{C}^k(\mathbb{R}, \mathbb{C}^{n\times n})$ is unitary and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_n) \in \mathcal{C}^k(\mathbb{R}, \mathbb{R}^{n\times n})$, as the unique solution of the following system of differential equations ([4, eqs. (2.6–2.7)]):

$$\begin{aligned}
&\dot{U} = UH\ ,\\
&\dot{\lambda}_j = (U^*A_\gamma U)_{jj}\ ,\ j = 1, \dots, n,\\
&U(0) = U_0\ ,\ \Lambda(0) = \Lambda_0,\\
&\qquad \text{where } H \text{ is the skew-Hermitian function such that}\\
&H_{kj} = -\bar{H_{jk}} = (U^*\dot{A}_\gamma U)_{kj}/(\lambda_j - \lambda_k)\ ,\quad k < j,\\
&H_{jj} = 0\ ,\ j = 1, \dots, n.
\end{aligned} \tag{2.1}$$

As proved in [8, Theorem 2.6], the solution U gives the same Schur form as the "minimum variation decomposition" (MVD) of [3] and—at the end of the loop γ—we will have

$$U(1) = U(0)\Phi \quad \text{with } \Phi = \mathrm{diag}\big(e^{i\alpha_j}\ , j = 1, \dots, n\big)\ ,\ \ -\pi < \alpha_j \leq \pi\ , j = 1, \dots, n, \tag{2.2}$$

where the α_j's are the *Berry phases* associated with the loop γ. We call this Schur decomposition the MVD of A_γ. Note that we are taking α_j's to be the principal logarithms of the diagonal entries of $U(0)^*U(1)$.

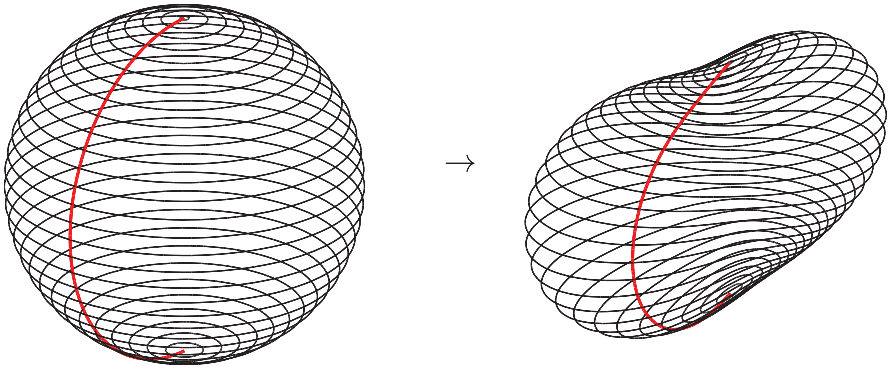

FIG. 1. *Covering on S inherited from the standard 2-sphere.*

2.2. Localization of coalescing points. Our algorithm localizes *generic* coalescing points of the function A (see [8] for definition of generic coalescing points). The mathematical basis of our algorithmic developments is given by the work [8], which provides a rigorous and thorough mathematical justification to a remarkable insight of Stone [15]. The basic idea is as follows.

Consider the region Ω, which we recall is diffeomorphic to the unit ball, and a covering for the surface bounding it. Borrowing terminology from the familiar one used for the sphere bounding the unit ball, we will say that the covering will go from the south to the north pole, and similarly we will talk about *parallels* and *meridians* on the surface bounding Ω. See Figure 1. Now, let us fix a meridian, and suppose we are moving from the south to the north pole along this specific meridian, covering the surface through its parallels (which are of course closed loops). Let $s \in [0,1]$ be the parameter used for the meridian ($s = 0, 1$ corresponds to south and north poles, respectively), so that each parallel is identified by the value s of the meridian. Along each of these parallels, we compute *smooth Berry phases* relative to the n different eigenvalues. The wording "smooth," in this context, signifies that the $\alpha_j(s)$'s in (2.2) are chosen to be smooth functions of the parameter s used for the meridians. This implies that they no longer necessarily take values in $(-\pi, \pi]$, but are allowed to move into adjacent branches of the logarithm function to retain continuity.

Finally, relative to these functions $\alpha_j(s)$'s, $0 \le s \le 1$, with $\alpha_j(0) = 0$, $j = 1, \dots, n$, the surface is declared *phase preserving* for A if

$$\alpha_j(1) = 0 \ \forall \ j = 1, \dots, n,$$

and it is called *phase rotating* if it is not phase preserving (see [15]).

Now, let $A \in \mathcal{C}^k(\Omega, \mathbb{C}^{n\times n})$ be Hermitian, let S be the surface bounding Ω, and suppose there are no coalescing eigenvalues for A on S. A complete justification of the following result (which in essence is the original insight of Stone) can be found in [8].

THEOREM 2.1 (Stone, [15]). *If S is phase rotating, then there is a coalescing point for A in Ω (i.e., inside S).* □

We are now ready for the theoretical results which form the backbone of our algorithm for the localization of coalescing points. Statement (2.3) is Theorem 4.10 in [8], while the remaining part of Theorem 2.2 below is Theorem 4.13 of [8], to which we refer for details.

THEOREM 2.2. *Let $A \in \mathcal{C}^k(\Omega, \mathbb{C}^{n\times n})$, $k \geq 1$, be Hermitian. Let $\lambda_1(x), \dots, \lambda_n(x)$ be its continuous eigenvalues, labeled in descending order. Let S be the surface bounding Ω and assume that there are no coalescing points on S (distinct eigenvalues on S).*

For each $j = 1, \dots, n$, let $\alpha_j(s)$ be the Berry phase function associated with λ_j. Then we have that

$$\sum_{j=1}^{n} \alpha_j(s) = 0 \quad \textit{for all } s \in [0,1]\,. \tag{2.3}$$

Moreover, suppose that, for any $j = 1, \dots, n-1$, we have

$$\lambda_j = \lambda_{j+1}$$

precisely at d_j distinct generic coalescing points in Ω, and that there is a total of $N = \sum_{j=1}^{n-1} d_j$ distinct coalescing points for A in Ω. Then we have that

$$\begin{array}{llll} \text{(i)} & \alpha_1(1) = 0 \ (\textit{resp., } 2\pi) & (\text{mod } 4\pi) & \textit{if} \ \ d_1 \textit{ is even } (\textit{resp., odd}), \\ \text{(ii)} & \alpha_j(1) = 0 \ (\textit{resp., } 2\pi) & (\text{mod } 4\pi) & \textit{if} \ \ d_{j-1} + d_j \textit{ is even } (\textit{resp., odd}) \\ & & & \quad \textit{and } j = 2, \dots, n-1, \\ \text{(iii)} & \alpha_n(1) = 0 \ (\textit{resp., } 2\pi) & (\text{mod } 4\pi) & \textit{if} \ \ d_{n-1} \textit{ is even } (\textit{resp., odd}) \end{array}$$

with $|\alpha_1(1)| \leq 2d_1\pi$, $|\alpha_n(1)| \leq 2d_{n-1}\pi$, and $|\alpha_j(1)| \leq 2(d_{j-1} + d_j)\pi$ for $j = 2, \dots, n-1$. □

The next theorem (again, see [8]) goes in the converse direction of Theorem 2.2 just stated, and provides a practical criterion under which a coalescing of eigenvalues takes place. See also Remark 2.5 for more refined criteria.

THEOREM 2.3. *Let A, Ω, and S be just as in Theorem* 2.2. *If there exists an index k, $1 \leq k \leq n$, for which $\alpha_k(1) \neq 0$, then λ_k must have coalesced at least once inside Ω with λ_{k-1} or λ_{k+1}.* □

Remark 2.4. Theorem 2.3 is a topological result similar in spirit to the intermediate value theorem (IVT) of calculus. Indeed, much like the IVT lends the theoretical support for powerful localization techniques of roots of scalar equations (e.g., bisection), Theorem 2.3 will provide the theoretical support to our localization technique for coalescing points. And, just as bisection may fail to locate multiple (hence, nongeneric) roots, our method also will generally be able to localize only generic coalescing points.

Remark 2.5. The following refinement of Theorem 2.3 is immediate.

- Refinement of Theorem 2.3. *Let p, q, with $1 \leq p < q \leq n$ be such that*
  (a) $\sum_{j=1}^{p-1} \alpha_j(1) = 0$ *and* $\alpha_p(1) \neq 0$,
  (b) *q is the smallest index for which $\sum_{j=p}^{q} \alpha_j(1) = 0$.*
  Then, $(\lambda_p, \lambda_{p+1}), \dots, (\lambda_{q-1}, \lambda_q)$ have coalesced at least once inside Ω. □

2.3. 1-d continuation: A predictor–corrector method. As made clear in the previous section, we need to approximate the MVD decomposition of A along parallels, that is the solution of (2.1), and this must be done accurately so that the information on the Berry phases is reliable. In this section, we propose a new predictor-corrector method to do this.

We remark (see [8, Theorem 2.4]) that along each parallel we can use any 1-periodic parametrization, call t the parameter used along each parallel, and denote by $B(t)$, $t \in [0,1]$, the restriction of the function A to the parallel corresponding to the value s of the meridian.

Below, we assume we have a partition of the interval $[0,1]$ which describes the mesh for a given parallel (in practice, this mesh will be found adaptively; see later):

$$0 = t_0 < t_1 < \cdots < t_N = 1 \ , \ \ h_k = t_{k+1} - t_k \ , \ \ k = 0, 1, \ldots, N-1 \ . \tag{2.4}$$

Predictor-Corrector Method. Given $B \in \mathcal{C}_1^p(\mathbb{R}, \mathbb{C}^{n\times n})$, we seek a MVD of $B(t)$, $U^*(t)B(t)U(t) = \Lambda(t)$, at $t = t_0, t_1, \ldots, t_N$. Call U_k the computed approximations to $U(t_k)$.

0. (*Initialization*). At $t_0 = 0$, let $B(0) = U_0\Lambda(0)U_0^*$ be an ordered Schur factorization, that is $U(0) = U_0$.

1. For $k = 1, \ldots, N$:

- (*Predictor phase*). At t_k, let $B(t_k) = Q_k\Lambda(t_k)Q_k^*$, where Q_k is unitary and $\Lambda(t_k)$ is the real diagonal matrix of the ordered eigenvalues of $B(t_k)$.
- (*Corrector phase*). Compute a phase matrix Φ_k such that $\|U_{k-1} - Q_k\Phi_k\|_F$ is minimized, and set $U_k = Q_k\Phi_k$.

Before justifying our predictor-corrector method, it is important to point out the two key difficulties in the task at hand.

(1) It is hard to imagine being able to obtain a very accurate approximation to the MVD of A at the t_k's along the given parallel in a way that is less costly than that of computing a Schur decomposition at the t_k's.

(2) It is hopeless to expect being able to find the solution of (2.1), U, at the mesh points t_k's, just by computing a Schur factorization at the t_k's. In fact, the severe lack of uniqueness in the Schur factorization makes it impossible to retrieve the information on the phase matrix (which is the key quantity we need) purely from a Schur factorization at the t_k's.

Of course, we could have used any integration technique on (2.1). In fact, this could be done at an arbitrarily high order of accuracy and with a host of different methods which are further guaranteed to maintain unitary approximations. We made some experiments with this approach, but were ultimately disappointed, for two reasons. (a) The method, implemented in such a way to return unitary approximations, turned out to be very expensive, considerably more so than the expense of performing a Schur decomposition per step: given that we need to approximate the MVD of A along *many* parallels,[1] this appears undesirable. (b) It was hard to obtain a good approximation to the Berry phases.

Our method is such that, see Theorem 2.7, (i) at leading order of expense, it costs as much as one Schur decomposition per step, (ii) it requires no integration of (2.1), and (iii) it is guaranteed to have 2nd order of accuracy globally, also for the Berry phases.

Remark 2.6. The idea of the method is similar to one we had adopted in [5] to compute a smooth Schur decomposition of a one-parameter real *symmetric* matrix valued function. The key difference is that—in that context—we were able to retrieve the exact factor $U(t_k)$. This was because any algebraic factor only required adjustments of the signs of its columns (in other words, the phase matrix was simply $\mathrm{diag}(\pm 1)$). Now, this is no longer possible.

Theorem 2.7. *Let $A \in \mathcal{C}^p(\Omega, \mathbb{C}^{n\times n})$, $p \geq 3$, be a Hermitian matrix valued function. Let S be the surface bounding Ω, and assume that A has no coalescing eigenvalues on S. Let s be a given value of the meridian, and let $B \in \mathcal{C}_1^p(\mathbb{R}, \mathbb{C}^{n\times n})$,*

[1] In principle, we would need to find the MVD of A along all parallels, that is along each parallel for all values of s along the meridian, but in practice we use an adaptive technique choosing the meridian values where the MVD along parallels is actually approximated.

$p \geq 3$, be the restriction of the function A to the parallel corresponding to the value s of the meridian.

Let a partition (2.4) be given, with $h = \max_k h_k$ sufficiently small, and let U_k be the approximation to $U(t_k)$ found from the above predictor-corrector method, for $k = 1, 2, \dots, N$.

Then, the factor U_N is a second order (i.e., $\mathcal{O}(h^2)$) accurate approximation to the exact factor $U(1)$. As a consequence, the approximate phase matrix $U_N^* U_0 = U_N^* U(0)$ (which is diagonal) is a second order accurate approximation to the exact phase matrix $U(1)^* U(0)$, and the computed Berry phases $\alpha_j(s)$, $j = 1, \dots, n$, are also second order accurate, for any $s \in [0, 1]$.

Proof. First of all, we notice that the constrained minimization problem $\min_{\Phi_k} \|U_{k-1} - Q_k \Phi_k\|_F$, subject to the constraint that Φ_k be a phase matrix, is solved by the choice

$$\Phi_k = \exp(i \operatorname{Arg}(D_k)), \quad D_k = \operatorname{diag}(Q_k^* U_{k-1}),$$

where Arg is the principal argument of a complex number.

Next, since the eigenvalues are distinct in $[0, 1]$, we have that both Q_k and U_{k-1} are within a phase factor of the exact factors $U(t_k)$ and $U(t_{k-1})$, respectively: $U_{k-1} = U(t_{k-1})\Psi_{k-1}$ and $Q_k = U(t_k)\Psi$, for some phase matrices Ψ_{k-1} and Ψ. Therefore

$$\begin{aligned} \Phi_k &= \exp\bigl[i \operatorname{Arg}\bigl(\operatorname{diag}\bigl(\Psi^* \bigl(U^*(t_k) U(t_{k-1})\bigr) \Psi_{k-1}\bigr)\bigr)\bigr] \\ &= \Psi^* \, \exp\bigl[i \operatorname{Arg}\bigl(\operatorname{diag}(U^*(t_k) U(t_{k-1}))\bigr)\bigr] \, \Psi_{k-1}. \end{aligned} \tag{2.5}$$

Now, let u_j be the jth column of U, so that

$$\operatorname{diag}(U^*(t_k) U(t_{k-1})) = \operatorname{diag}(u_j^*(t_k) u_j(t_{k-1}), j = 1, \dots, n).$$

Use the expansion $u_j(t_k) = u_j(t_{k-1}) + h_{k-1}\dot{u}_j(t_{k-1}) + \frac{h_{k-1}^2}{2}\ddot{u}_j(t_{k-1}) + \frac{h_{k-1}^3}{6}\dddot{u}_j(\tau_{k-1})$, where $\tau_{k-1} \in [t_{k-1}, t_k]$, and recall that $\dot{u}_j^*(t) u_j(t) = 0$. As a consequence, we get

$$u_j^*(t_k) u_j(t_{k-1}) = 1 + \frac{h_{k-1}^2}{2} \ddot{u}_j^*(t_{k-1}) u_j(t_{k-1}) + \mathcal{O}(h_{k-1}^3), \; j = 1, \dots, n\,. \tag{2.6}$$

Now, recall (see (2.1)) that $\dot{U} = UH$, and $H_{jj} \equiv 0$, so that $\ddot{u}_j^*(t_{k-1}) u_j(t_{k-1}) = (H^2)_{jj}(t_{k-1})$. But, we now observe that $H^2 = -H^* H$ is a Hermitian matrix, and thus the diagonal is real. Therefore, see (2.5)–(2.6), we need to find the value of Arg for a complex number of the form $\zeta = 1 + a h_{k-1}^2 + i(b h_{k-1}^3) + \cdots$, for which we immediately have that $\operatorname{Arg}(\zeta) = \mathcal{O}(h_{k-1}^3)$. Then, we readily have that

$$\bigl[i \operatorname{Arg}\bigl(\operatorname{diag}(U^*(t_k) U(t_{k-1}))\bigr)\bigr] = \mathcal{O}(h_{k-1}^3)$$

and thus

$$\Phi_k = \Psi^* \Psi_{k-1} (I + \mathcal{O}(h_{k-1}^3)),$$

and further

$$U_k = Q_k \Phi_k = U(t_k) \Psi \Phi_k = U(t_k) \Psi_{k-1} (I + \mathcal{O}(h_{k-1}^3)). \tag{2.7}$$

Since $U_k = U(t_k)\Psi_k$, and $\Psi_0 = I$, then for all $k = 1, 2, \dots,$ we get

$$\Psi_k = \left(I + \frac{h_0^3}{6} E_0\right) \left(I + \frac{h_1^3}{6} E_1\right) \cdots \left(I + \frac{h_{k-1}^3}{6} E_{k-1}\right) + \text{h.o.t.},$$

where h.o.t. indicates higher order terms, and the matrices $E_0, \dots, E_{k-1}$ comprise the leading terms in the local error expansion. From this, we obtain

$$U(t_k) - U_k = U(t_k)\big[I - \Psi_k\big] = -U(t_k)\sum_{j=0}^{k-1}\frac{h_j^3}{6}E_j + \text{h.o.t.} \tag{2.8}$$

Finally, since the error accumulates according to (2.8), the factor U_N is a second order accurate approximation to the exact factor $U(1)$.

That $U_0^*U_N = U(0)^*U_N$ is diagonal is simply because U_N differs from $U(0)$ by a phase matrix, and that $U_0^*U_N$ is a second order accurate approximation to the exact phase matrix $U(0)^*U(1)$ is an immediate consequence of the fact that U_N is a second order accurate approximation of $U(1)$. Finally, the statement on the Berry phases is a consequence of the fact that $U_0^*U_N = U_0^*U(1)U^*(1)U_N = \Phi\big(I + \text{diag}(\mathcal{O}(h^2))\big)$ and that, for $|z| < 1$, $\log(1+z) = z - z^2/2 + \cdots$. □

Remark 2.8. A couple of observations are in order.

1. The result in Theorem 2.7 that the global error is of second order accuracy appears surprising. To appreciate this statement, consider what we do at the first mesh point t_1. Clearly, $U(t_1) = U_0 + \mathcal{O}(h_0)$, and so U_0 (which is what we use in order to correct *some* Schur factorization Q_1) is only a first order approximation to the exact factor at t_1 and a standard error analysis would then give us a zeroth order method. However, our algorithm behaves like a second order method used to integrate (2.1), in spite of the fact that no integration of the differential equation is actually performed.
2. The precise constant hidden in the $\mathcal{O}(h_{k-1}^3)$ term depends on the third derivative of U. This suggests a strategy of adaptive time stepping along parallels in which the error per step is kept constant.

3. Algorithms.

The techniques described in section 2 have been implemented in a MATLAB code and we will report on several experiments in section 4. Here we give specific details of our implementation choices.

The first choice we made is to work with *cubes*; see Figure 2. Naturally, any surface S bounding a region Ω diffeomorphic to the unit ball can be mapped homeomorphically into a cube. But, the cube has corners, and in particular parallels are now squares and we will need to integrate along piecewise smooth curves, not smooth curves. However, this has no theoretical nor practical consequences, since the location of the corners is known a priori and in our algorithm we will always require stepping exactly at these values where the curve is not differentiable with no loss of order of accuracy.

With this in mind, we will cover a cube proceeding from the south pole (the baricenter of the bottom square) to the north pole, by integrating along squares (the parallels) on the three different regions of the cube: bottom, lateral, top; again, see Figure 2.

3.1. Step size selection.

Here we discuss the adaptive step size algorithms we implemented, both along parallels, and along the meridian (south-to-north pole).

- *Along parallels.* Along a parallel, that is to cover a square of side L, our step size selection is based on a standard error per step control, whereby we attempt to keep the error on each step constant. In the notation used to describe our predictor-corrector method, we did the following:
  (a) compute $\rho_U = \|U_k - U_{k-1}\|_1$ and $\rho_\Lambda = \max_j \frac{|\lambda_j(t_k) - \lambda_j(t_{k-1})|}{1+|\lambda_j(t_k)|}$;

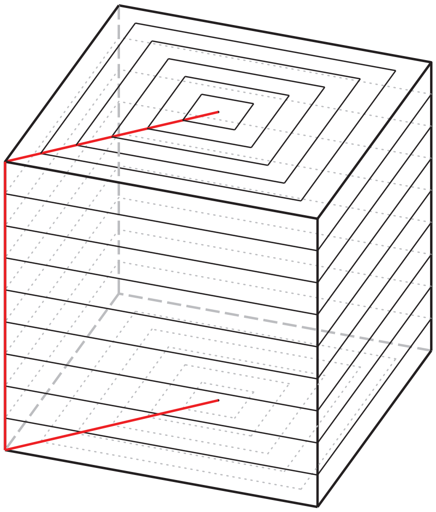

FIG. 2. *Covering of the cube.*

(b) let $\rho = \max(\rho_U, \rho_\Lambda)/\texttt{tolls}$ and $h_{\text{new}} = \min(h_{\max}, h_{k-1}/\rho)$;

(c) if $\rho \leq 1.2$, the step is accepted, otherwise it is not. In either case, the new step size is given by h_{new}, and the computation is continued until either $h_{\text{new}} < h_{\min}$ or we reached the last point.

For reference, in our experiments we used $\texttt{tolls} = 10^{-1}$, $h_{\max} = L/10$, $h_{\min} = 10^{-14}$. Finally, we always adjust the step size (i.e., restrict it as needed) in order to step exactly at all corners of the square we are covering. Along the very first parallel, we use $h_0 = h_{\max}$; all other times we use the last step size which we would have used along the previous parallel without accounting for the artificial shortening of it in order to step at the last point. Similarly along a given parallel when we need to restart from the corners of the current square.

- *Stepping along the meridian.* Choosing the step size to move along the selected meridian is more delicate. The key concern now is to make sure that the Berry phases are accurately approximated, in particular that a fast variation of the same is not missed. The goal, here, is to try to ensure that we do not jump from a branch of the logarithm to an adjacent one (a fact, which ultimately could happen if—some of—the Berry phase functions α_j's have a large first derivative; see Example 3.3).

Below, let

$$s_0 = 0 < s_1 < s_2 \cdots < s_K = 1$$

be the mesh (we will need to find this) along the meridian, and $\Delta_k = s_{k+1} - s_k$, $k = 0, 1, \ldots, K-1$. Also, let α be the n-vector of the (approximate) Berry phases, $\alpha : [0,1] \to \mathbb{R}^n$, and finally let $\texttt{tollp}$ be a fixed tolerance value to monitor variation in the phases. The strategy we used is as follows:

(a) compute $\delta\alpha = ||\alpha(s_k + \Delta_k) - \alpha(s_k)||_\infty/\texttt{tollp}$;

(b) let $\Delta_{\text{new}} = \min(2\Delta_k, \Delta_{\max}, \Delta_k/\delta\alpha)$;

(c) if $\delta\alpha \leq 1.5$, the step is accepted, otherwise it is not. In either case, the new step size is given by Δ_{new}, and computation is continued until either the step is successful or $\Delta_{\text{new}} < \Delta_{\min}$.

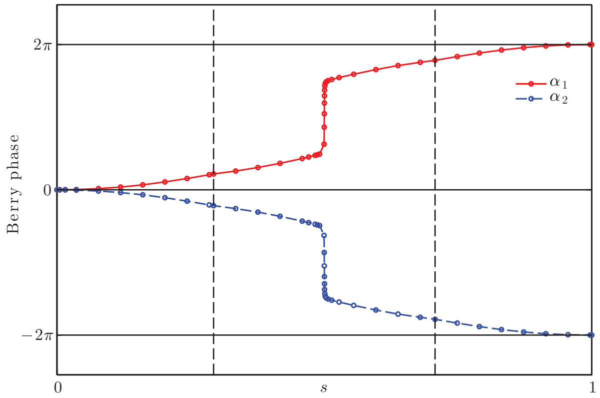

FIG. 3. *Example* 3.2, $k = 3$.

> In our experiment we used `tollp` $= \pi/6$, which corresponds to restricting a phase variation to be at most $\pi/4$. Further, we used $\Delta_{\min} = 10^{-14}$ and $\Delta_0 = \Delta_{\max}/8$, with $\Delta_{\max} = L/10$, along the meridian of a cube of side L. An extra cautionary control is performed to ensure that (2.3) is not violated and that we are properly following the *smooth* Berry phases; in detail, we reject a step, and halve it, if $|\sum_j(\alpha_j(s_{k+1}) - \alpha_j(s_k))| >$ `tolls`, where `tolls` is the tolerance used for error control along the parallels. Finally, the values of the step size are adjusted so to make sure that we step exactly at the corners of the red "C"-shaped curve in Figure 2 and the last step is also typically adjusted in order to have $s_K = 1$.

When we have completed covering the cube from the south to the north poles, we monitor the Berry phases and use Theorem 2.3 to infer which pairs of eigenvalues (if any) have coalesced inside the current cube.

Remark 3.1. It must be stressed that the default values of the tolerances we used are empirical, albeit based on sound underlying principles and extensive experimentation. That said, as is the case for any algorithm where the step size is chosen adaptively, it is typically possible to find pathological examples where one may obtain erroneous results; see Example 3.3.

Example 3.2. This example is chosen to illustrate that it is computationally demanding to integrate *near* a coalescing point because of the rapid change in the geometric phase and it is mandatory to use the adaptive step size strategy described above. The function is the simple function $A(x, y, z) = \left[\begin{smallmatrix} x & y+iz \\ y-iz & -x \end{smallmatrix}\right]$, which has a unique coalescing point for the eigenvalues at the origin (a generic coalescing point). The region of interest is $\Omega = [-10^{-k}, 2 - 10^{-k}] \times [-1, 1]^2$, $k = 0, \ldots, 3$. Ω is chosen so that the distance between the coalescing point and the boundary of Ω is 10^{-k}.

For all values of $k = 0, 1, 2, 3$, our computations give $\alpha_1(1) = 2\pi$, $\alpha_2(1) = -2\pi$, betraying the coalescing point. However, in the case of $k = 0$, integration along the cube proceeds swiftly with constant step sizes equal to $h_{\max}$ and $\Delta_{\max}$, whereas in the case of $k = 3$ the step sizes along the central side of the meridian become three orders of magnitude smaller; to witness, for $k = 3$, the minimum step size along the meridian is 1.14×10^{-4} and along the parallels is 1.86×10^{-4}. In Figure 3 we show $\alpha_{1,2}$ for the case $k = 3$; the markers on the curves are the meridian's steps.

Example 3.3. This example is chosen to illustrate that it is important to carefully monitor variation of the Berry phases when choosing the step length along the meridian; otherwise, one may end up on the wrong branch.

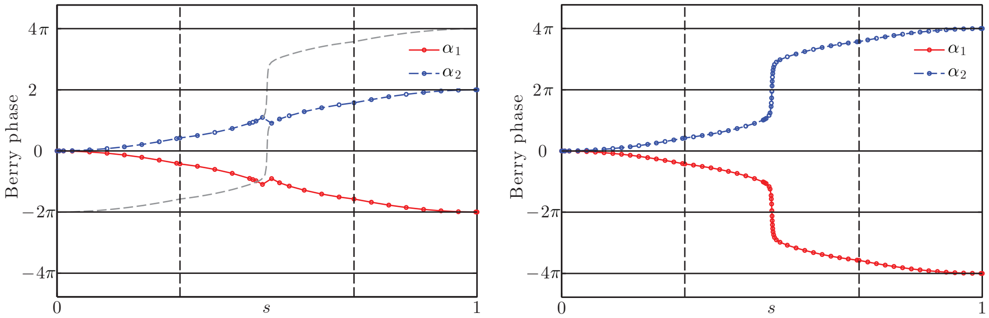

FIG. 4. *Example* 3.3. *Left,* $\Delta_{\max} = L/10$*; right,* $\Delta_{\max} = L/20$.

Consider the function $A(x, y, z) = \left[\begin{smallmatrix} xy & x^2-y^2+c^2+iz \\ x^2-y^2+c^2-iz & -xy \end{smallmatrix}\right]$, with $c = 1/100$, and the cube $\Omega = [-0.1, 1.9] \times [-1, 1]^2$. There are two coalescing points in the cube, $(0, c, 0)$ and $(0, -c, 0)$, both generic. In Figure 4, we show results obtained with $\Delta_{\max} = L/10$ (our default choice) and the more stringent value $\Delta_{\max} = L/20$; here, $L = 2$. In the first case, we fail to follow the correct branch and jump on the branches coming from $\pm 2\pi$; on the left of Figure 4, we show the dashed line of the phase coming from -2π, which is clearly getting close to the branch of α_2 we are following, and the algorithm switches—discontinuously—branches. In the second case (on the right of Figure 4), we follow the correct branches and approach the values $\pm 4\pi$, as we should. Note that the plot on the left would give the wrong information that only one coalescing point occurs inside the cube.

3.2. Refinement. A lofty goal of our computations is to reach a high degree of confidence that within a certain cube there are *no generic coalescing points* for a certain pair of eigenvalues (see Remark 2.4). Naturally, this must be understood within the realm of the limitations imposed by the task itself, in the sense that no region, no matter how small, can be guaranteed not to contain conical intersections (CIs for short), simply because one may have situations where two such CIs are close to each other and their combined effect make them unnoticed to the method. Nevertheless, a topological statement like Theorem 2.3 lends itself to an interpretation whereby within a certain cube we are able to exclude generic coalescing points. In practical terms, this type of conclusion would be all the more useful the larger the cube, while at the same time the larger the cube is the more likely it is that we have had a pair of eigenvalues coalescing repeatedly having gone unnoticed. This line of reasoning makes it necessary to *refine* a cube in smaller and smaller cubes, until either a pair of coalescing eigenvalues is isolated or the cube is so small that we can declare that no generic coalescing occurs in the cube within the limitations imposed by the computational power we have (or that we are willing to invest).

The basic refinement strategy we implemented is a straightforward subdivision of a cube into several smaller cubes, obtained by subdividing the edge of the original cube into N equal edges. This will give us N^3 equal cubes of edge L/N. Now we will have to compute the eigendecompositions (and monitor Berry phases) on smaller cubes, each two of which share a face in common, namely, the top face of a cube is the bottom face of the cube stacked above it. To save on computational effort, we can (and do) reuse the computations done on the top face and do not repeat them for the bottom face of the cube above it. Besides this global refinement strategy, in order to locate more accurately a coalescing pair of eigenvalues, we also implemented

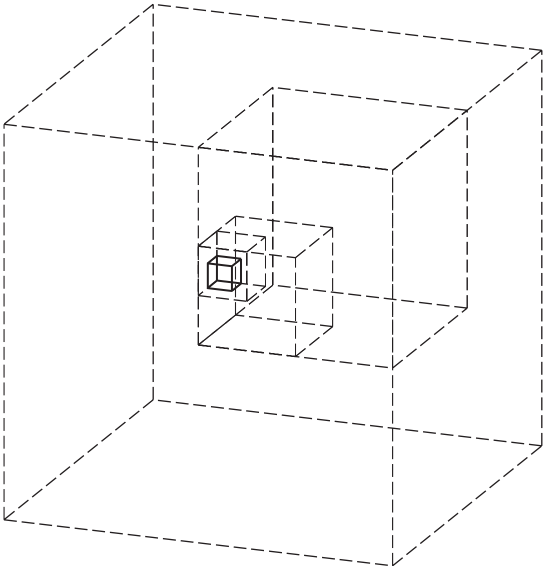

FIG. 5. *Local refinement isolating a coalescing point.*

a recursive refinement scheme progressively subdividing just a subcube, inside which a coalescing point has been detected, into eight equal cubes, obtained by bisecting the edges of the cube. By repeated application of this simple refinement procedure on a cube of edge L, after N subdivision steps we will have obtained a small cube of edge $L/2^N$ inside which we have isolated a coalescing point. See Figure 5.

3.3. Zoom in. Newton's method with coordinate search. Once we have localized a cube (say, of edge L) inside which we know that there is a coalescing pair of eigenvalues (say, λ_k and λ_{k+1}), we may want to locate accurately the value of x where the coalescing occurs. As an alternative to progressive refinement of the cube as above, we used (see also [7]) Newton's method to minimize the nonlinear smooth functional

$$f(x) = (\lambda_k(x) - \lambda_{k+1}(x))^2 ,$$

that is, to solve $F(x) \equiv \nabla f(x) = 0, \quad F : \mathbb{R}^3 \to \mathbb{R}^3$. On this system, we perform Newton's method by working with standard centered differences approximations for the first and second derivatives which are needed to form F and the Jacobian DF. Namely, taking h equal to the cubic root of the machine precision `eps`, we use

$$F_j(x) = f_{x_j}(x) \approx \big(f(x+e_jh) - f(x-e_jh)\big)/(2h) \ , \ j = 1,2,3,$$

$$(DF)_{jj}(x) = f_{x_jx_j}(x) \approx \big(f(x+e_jh) - 2f(x) + f(x-e_jh)\big)/h^2 \ , \ j = 1,2,3,$$

and

$$\text{For} \quad i,j = 1,2,3,\, i \neq j \ , \qquad (DF)_{ij}(x) = f_{x_i,x_j}(x) \approx \frac{f(x+e_ih+e_jh) - f(x+e_ih-e_jh) - f(x-e_ih+e_jh) + f(x-e_ih-e_jh)}{4h^2} \ .$$

Before proceeding, we need to justify the use of Newton's method, in particular that the function F above is a smooth ($\mathcal{C}^2$) function, as otherwise Newton's method may exhibit erratic convergence behavior (if at all). Note that at first glance it is not clear that F is smooth at all, since the eigenvalues λ_k and λ_{k+1} are merely continuous at a coalescing point ξ (since ξ is a conical intersection point). However, as shown in the following lemma, the function f is as smooth as the original function A.

LEMMA 3.4. *Let $A \in \mathcal{C}^p(\Omega, \mathbb{C}^{n\times n})$, $p \geq 3$, be Hermitian. Let $\lambda_1 \geq \cdots \geq \lambda_n$ be its ordered eigenvalues, and let ξ be a generic coalescing point for the pair of eigenvalues λ_k and λ_{k+1}, with no other pair of eigenvalues coalescing at ξ. Let ρ be the distance from ξ to the nearest point in Ω where another pair of eigenvalues coalesce and let $B_\rho(\xi)$ be the open ball centered at ξ, and of radius ρ (we formally set $B_\rho(\xi) = \mathbb{R}^3$ if no further coalescing occurs in Ω). Finally, let $D = B_\rho(\xi) \cap \Omega$. Then, the function f, defined as $f(x) = (\lambda_k(x) - \lambda_{k+1}(x))^2$, is $\mathcal{C}^p$ in D. In particular, $F := \nabla f$ is $\mathcal{C}^{p-1}$ in D.*

Proof. Throughout, restrict to D. If $j \neq k, k+1$, the eigenvalues λ_j's of A are distinct in D, and hence (see Lemma 1.1) in D they are as smooth as A. Next, consider the characteristic polynomial of A, $p(A)$, and its discriminant $d(A)$ (e.g., see [1]). Of course, $p(A)$ has coefficients as smooth as A, which also happen to be real (because the eigenvalues of A are real and $p(A) = (\lambda - \lambda_1)\cdots(\lambda - \lambda_n)$). Now, it is known that $d(A)$ is a homogeneous polynomial of degree $2(n-1)$ in the coefficients of the characteristic polynomial of A, and thus it follows that $d(A)$ is as smooth as A. It is also known that $d(A)$ vanishes only at repeated roots of the characteristic polynomial, that is, in our case just at ξ. Let us write $d(A)$ (this is often taken as the definition of $d(A)$) in terms of the roots of the characteristic polynomial as

$$d(A) = \prod_{i<j} (\lambda_i - \lambda_j)^2 \ .$$

From this last expression, we isolate the term $(\lambda_k - \lambda_{k+1})^2$, and call $g(\lambda_1, \ldots, \lambda_n) = \frac{\prod_{i<j}(\lambda_i - \lambda_j)^2}{(\lambda_k - \lambda_{k+1})^2}$. So doing, we get

$$(\lambda_k - \lambda_{k+1})^2 = d(A)/g(\lambda_1, \ldots, \lambda_n),$$

where, in D, numerator and denominator are as smooth as A, and $g(\lambda_1, \ldots, \lambda_n)$ does not vanish. □

Remark 3.5. Finally, the following remarks are important in order to evaluate the practical performance of the present zoom-in module.

(i) As it turns out, it is very important to choose the starting point for the Newton's iteration carefully. At first, we started Newton's method with an initial guess located at the baricenter of the cube, but we often failed to converge within the allowed max number of iterations (practically, 10), or ended up outside of the cube. To find an improved starting value for Newton's iteration, and avoid unduly refining the cube, we used a classical coordinate search method, which we now briefly describe.

(a) Let $x^{(0)}$ be the baricenter of the cube, and let $\delta = L/4$.

(b) Repeat *until convergence*

- find a trial point $x_t \in \{x^{(0)} \pm e_i\delta,\ i = 1, 2, 3\}$ such that $f(x_t) < f(x^{(0)}) - 10^{-4}\delta^2$ (*decrease condition*);
- if none of the trial points satisfies the decrease condition, then set $\delta = \delta/2$ otherwise, set $x^{(0)} = x_t$.

The method, which is known to be globally, though slowly, convergent, can be easily extended to our bound constrained minimization problem by suitably projecting on the cube's surface the trial points which end up outside of the cube (see [11]). We halted the procedure when $\delta < \delta_{\min} = 10^{-3}L$, and used the last reached value of $x^{(0)}$ as the starting point for Newton's method. This enormously enhanced the convergence properties of Newton's method, and ended up significantly decreasing convergence failures for it, even for cubes of relatively large side; see Example 4.1.

(ii) It should be appreciated that Newton's method (and coordinate search as well) is rather expensive, since one eigendecomposition is needed for each f-evaluation, and 19 eigendecompositions are required in order to evaluate the function F and its Jacobian. Of course, we could use some modified Newton iteration, but we did not attempt this. Yet another possibility is to refine the cube. However, the latter approach requires integration on smaller and smaller cubes which have a coalescing point inside. As remarked in section 3.1, integration along (small) loops near a coalescing point requires very small step sizes and it is an expensive and delicate task; indeed, in our experience, it is less expensive to trigger a zoom-in process as the one described above, when a coalescing point has been isolated and we want to locate it at high accuracy.

(iii) Of course, using divided difference approximations—and accounting for both truncation and roundoff errors—does put some limitation on the overall accuracy we can reach. For our specific choices, we will be bound to have an overall error of $\mathcal{O}(\texttt{eps}^{3/2})$, that is we should not expect errors smaller than roughly 10^{-12}. This bound was indeed observed in our computational experiments.

3.4. Parallelization. Our algorithm is inherently *geometrically* parallelizable, and it scales very favorably with respect to the number of processors. For example, consider the previously described global subdivision scheme, in which we subdivided an edge of length 1 into N equal segments. Thus, we will have N^2 columns each made up of N cubes. These columns do not need to communicate with one another, and hence the decomposition along each of them can be distributed to a different processor.

We have implemented this parallelization strategy on the distributed computing environment `FoRCE` of Georgia Tech. This is a dedicated computer cluster, consisting of a total of nearly 15,000 CPU cores. The parallelization strategy has been essential to perform some of the computations in the next section, namely, those for Example 4.3. To illustrate, for that problem with $n = 20$ (the dimension of the matrices), we have used $N = 2^k$ processors, $k = 0, 1, \dots, 5$. With respect to the execution time (that is, elapsed time) of this computation on one single processor, distributing the computations across $2^k, k = 1, \dots, 5$, processors gave a reduction by a factor of $0.5, 0.26, 0.13, 0.067, 0.035$, respectively, reflecting a nearly ideal workload distribution. (For the sake of reference, the elapsed time for one processor exceeded 5 hours.)

3.5. Large problems: Following a subset of eigenvalues. For large problems (that is, for large values of n), it becomes unfeasible to compute (repeatedly) a full eigendecomposition of A. Fortunately, in many circumstances it is sufficient to find conical intersections of a small number of dominant eigenvalues, say for $\lambda_1 \geq \cdots \geq \lambda_p$, with $p \ll n$. In this case, one is interested in the eigenspace associated with these eigenvalues only. That is, the full eigendecomposition of A is now replaced by the restricted decomposition $AU = U\Lambda$, with $U \in \mathbb{C}^{n\times p}$ orthonormal, $U^*U = I_p$, and $\Lambda = \mathrm{diag}(\lambda_1, \dots, \lambda_p)$. In the case of distinct (and ordered) eigenvalues, the degree of uniqueness of this restricted eigendecomposition is now with respect to a (p, p) phase matrix; that is, $U \to U\Phi$, where $\Phi = \mathrm{diag}(e^{i\alpha_j},\ \alpha_j \in \mathbb{R},\ j = 1, \dots, p)$.

Both theory and algorithms are easily adapted to this case with minor modifications. The main modification is due to the fact that (2.3) is no longer necessarily true when the index n in the sum there is replaced by p, and therefore the refinement of Remark 2.5 is also not necessarily true in the stated forms. However, the concept of Berry phase holds unchanged (since it is a quantity associated with each eigenvector separately), and so does the concept of phase rotating/preserving surface relatively to this subset of p dominant eigenvectors.

The appropriate modification of Theorem 2.3, which is the result on which we base our detection technique, still holds true in the following form which we state without proof since it is immediate.

THEOREM 3.6. *Let $A \in \mathcal{C}^k(\Omega, \mathbb{C}^{n\times n})$, $k \geq 1$, be Hermitian. Let $\lambda_1(x) \geq \cdots \geq \lambda_n(x)$ be its continuous eigenvalues, labeled in descending order, and let $p < n$. Let S be the surface bounding Ω and assume that there are no coalescing points on S for $\lambda_1, \dots, \lambda_p$. If there exists an index k, $1 \leq k \leq p$, for which $\alpha_k(1) \neq 0$, then λ_k must have coalesced at least once inside Ω with λ_{k-1} or λ_{k+1}.* □

Note that if $k = p$ in Theorem 3.6, then—inside Ω—λ_p may have coalesced with λ_{p+1}, which we are not tracking. At the same time, λ_{p+1} may further have coalesced with λ_{p+2}, etc., but these possible coalescings will not be visible to us.

From the algorithmic point of view, the only technique that needs modification is the predictor-corrector algorithm, since now we seek a restricted eigendecomposition. These modifications are immediate, replacing every occurrence of the full eigendecomposition $A = U\Lambda U^*$ (which requires U to be unitary) with the invariant subspace relation $AU = U\Lambda$, which holds also in case U has fewer columns than rows (and is orthonormal). With similar obvious modifications, a result like Theorem 2.7 also continues to hold.

4. Numerical experiments.

We now show performance of the algorithm on problems of differing nature.

Example 4.1. In this problem, there are relatively few generic coalescing points in the given cube, and all aspects of the algorithm are tested: localization, refinement, and zoom in. The problem is given on the cube $\Omega = [0, 1]^3$, and computations are done with $\texttt{tolls} = 10^{-1}$, $\texttt{tollp} = \frac{\pi}{6}$, and zoom-in tolerances (convergence and eigenvalues' gap) both equal to 10^{-8}. The function is

$$A(x, y, z) = \left(1 - \frac{1}{2}x^2\right) H_1 + xH_2 + \left(1 - \frac{1}{2}y^2\right) H_3 + yH_4 + \left(1 - \frac{1}{2}z^2\right) H_5 + zH_6\,,$$

where

$$H_1 = \begin{bmatrix} -0.1 & -0.6+0.7i & 0.8+0.2i & -0.3+1.2i & -0.3+0.3i & -0.1+0.1i \\ -0.6-0.7i & -0.5 & -0.8+0.3i & -0.3-1.7i & 0.1-0.5i & -1+0.2i \\ 0.8-0.2i & -0.8-0.3i & 1.6 & 0.1-0.1i & 0.9 & 1.2-0.9i \\ -0.3-1.2i & -0.3+1.7i & 0.1+0.1i & 1.9 & -1.5-0.3i & 0.3i \\ -0.3-0.3i & 0.1+0.5i & 0.9 & -1.5+0.3i & 0.4 & 0.1-0.2i \\ -0.1-0.1i & -1-0.2i & 1.2+0.9i & -0.3i & 0.1+0.2i & -0.5 \end{bmatrix},$$

$$H_2 = \begin{bmatrix} -2.3 & 1.5+0.7i & -1.9 & 1-0.8i & 0.6+1.2i & -0.6-i \\ 1.5-0.7i & 1.5 & -0.1-0.1i & 0.4-0.3i & 0.3-0.5i & 0.3+0.5i \\ -1.9 & -0.1+0.1i & -1 & -0.2-1.4i & 0.9-0.3i & -0.1+0.9i \\ 1+0.8i & 0.4+0.3i & -0.2+1.4i & 0.3 & -0.4-0.2i & -0.3-i \\ 0.6-1.2i & 0.3+0.5i & 0.9+0.3i & -0.4+0.2i & -1.5 & 0.2-i \\ -0.6+i & 0.3-0.5i & -0.1-0.9i & -0.3+i & 0.2+i & 1 \end{bmatrix},$$

$$H_3 = \begin{bmatrix} 2.2 & 0.3-0.2i & 1.4+0.6i & -1.1+0.5i & 0.5-0.2i & -0.4-i \\ 0.3+0.2i & -1.9 & -0.9-1.1i & -0.5+0.8i & 0.3-0.5i & -0.1i \\ 1.4-0.6i & -0.9+1.1i & 0.6 & 0.6i & 0.2+0.5i & -2.1+0.3i \\ -1.1-0.5i & -0.5-0.8i & -0.6i & -0.5 & 0.7-0.5i & -0.2-0.6i \\ 0.5+0.2i & 0.3+0.5i & 0.2-0.5i & 0.7+0.5i & -0.3 & 0.3+0.9i \\ -0.4+i & 0.1i & -2.1-0.3i & -0.2+0.6i & 0.3-0.9i & 1.5 \end{bmatrix},$$

$$H_4 = \begin{bmatrix} 0.2 & 0.1i & -0.5+1.1i & 0.1-0.1i & 0.2-0.9i & -0.1+0.6i \\ -0.1i & -0.4 & -0.1+0.5i & 0.6+1.4i & -0.6+0.3i & 0.8-2i \\ -0.5-1.1i & -0.1-0.5i & -0.9 & -0.5-0.1i & -1-0.3i & 0.1+1.1i \\ 0.1+0.1i & 0.6-1.4i & -0.5+0.1i & 1 & 0.2-0.4i & 0.3+0.6i \\ 0.2+0.9i & -0.6-0.3i & -1+0.3i & 0.2+0.4i & 1.1 & -0.9+0.1i \\ -0.1-0.6i & 0.8+2i & 0.1-1.1i & 0.3-0.6i & -0.9-0.1i & -0.4 \end{bmatrix},$$

$$H_5 = \begin{bmatrix} -0.2 & -0.3-0.4i & 0.7-0.6i & -0.1-1.6i & -0.5+0.5i & -0.2-0.1i \\ -0.3+0.4i & -0.9 & 0.2-0.6i & -0.5+1.3i & 0.2i & -1.3-0.7i \\ 0.7+0.6i & 0.2+0.6i & 0.7 & 0.1+0.7i & 1-1.2i & -0.1 \\ -0.1+1.6i & -0.5-1.3i & 0.1-0.7i & 0.7 & -0.3+0.5i & 0.9+0.3i \\ -0.5-0.5i & -0.2i & 1+1.2i & -0.3-0.5i & 0.4 & -0.1+0.6i \\ -0.2+0.1i & -1.3+0.7i & -0.1 & 0.9-0.3i & -0.1-0.6i & 0.2 \end{bmatrix},$$

$$H_6 = \begin{bmatrix} 0 & 0.1-0.2i & 0.8-2.1i & 1+0.1i & 0.4+0.6i & -0.4-0.7i \\ 0.1+0.2i & 0.6 & -0.4-1.2i & -0.1+0.2i & -0.4-0.5i & 1.2-0.8i \\ 0.8+2.1i & -0.4+1.2i & -2 & 0.4-1.2i & -0.9i & 0.7i \\ 1-0.1i & -0.1-0.2i & 0.4+1.2i & 0 & -0.5-1.2i & -1.1-0.4i \\ 0.4-0.6i & -0.4+0.5i & 0.9i & -0.5+1.2i & -1.3 & 0.2-i \\ -0.4+0.7i & 1.2+0.8i & -0.7i & -1.1+0.4i & 0.2+i & -0.1 \end{bmatrix}.$$

Results of the simulation are displayed in Figure 6, from which we expect that λ_1 coalesces with λ_2, which coalesces with λ_3, and then λ_5 with λ_6. Although the

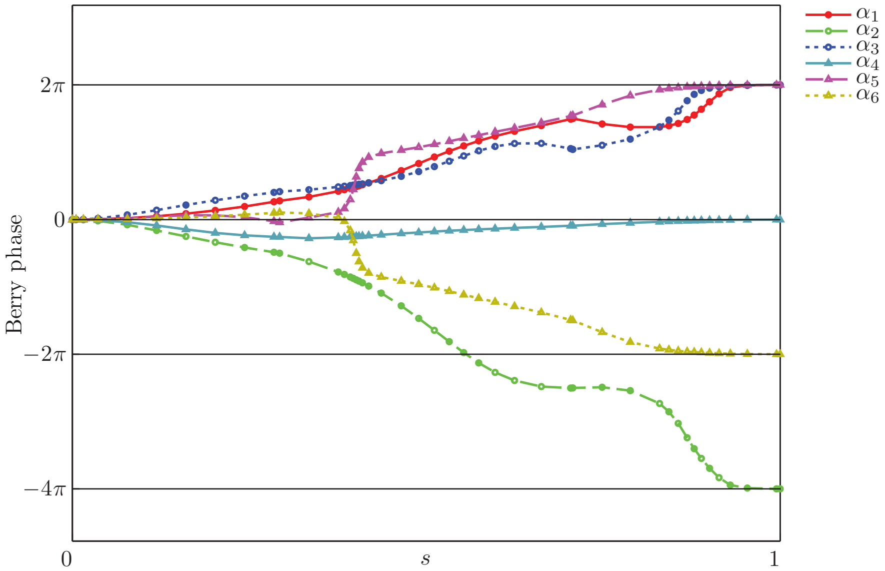

FIG. 6. *Example* 4.1.

Table 1
Zoom-in without coordinate search.

	$L_{\min}$	0.5	0.25	0.125
	# cubes	9	25	33
	s-steps/F	250/10	614/11	795/11
	t-steps/F	21686/965	38054/1190	45829/1244
$\lambda_1 = \lambda_2$	NIT	Fail	5	5
$\lambda_2 = \lambda_3$		Fail	Fail	5
$\lambda_5 = \lambda_6$		7	7	7

detection phase is already successful on the original cube, the refinement procedure is necessary in order to accurately locate (and isolate) the coalescing points. In detail, after refinement the code explored nine cubes, the original, and eight more resulting from one subdivision performed; all subcubes have edge of length $L = L_{\min} = 0.5$. Below, we report on the computed coalescing points, as well as on the number of coordinate search and Newton iterations performed by the zoom-in technique, denoted, respectively, by CSit and NIT.

- $\lambda_1 = \lambda_2$ at $\begin{bmatrix}0.44511899 & 0.34014156 & 0.94489258\end{bmatrix}$,
  CSit = 28 (visited 101 points), NIT = 3;
- $\lambda_2 = \lambda_3$ at $\begin{bmatrix}0.46761305 & 0.46167575 & 0.44946999\end{bmatrix}$,
  CSit = 23 (visited 83 points), NIT = 3;
- $\lambda_5 = \lambda_6$ at $\begin{bmatrix}0.80644491 & 0.87260280 & 0.41732847\end{bmatrix}$,
  CSit = 29 (visited 99 points), NIT = 4.

To contrast with the performance without coordinate search, we show in Table 1 the results obtained when we just start the Newton iteration at the baricenter of the current cube. $L_{\min}$ denotes the allowed minimum length of an edge; s-steps refers to the number of steps along the meridian, t-steps to the number of steps along the parallels, and F refers to failed steps along meridians and parallels. Without coordinate search, Newton's method often failed unless the cube was sufficiently small. In fact, two further subdivisions were needed in order to accurately find all the three coalescing points ($L_{\min} = 0.125$). Note also how this resulted in a considerable increment in the total number of steps taken along both parallels and meridians, with respect to the case $L_{\min} = 0.5$, which is sufficient for the success of the zoom-in procedure endowed with coordinate search.

Example 4.2. Here we show that our algorithm typically fails when used to find nongeneric coalescing points. Consider the functions:

$$A(x, y, z) = \begin{bmatrix} x^2 & y + iz \\ y - iz & -x^2 \end{bmatrix},$$
$$B(x, y, z) = \begin{bmatrix} x^2 + y^2 - r^2 & iz \\ -iz & -(x^2 + y^2 - r^2) \end{bmatrix}, r = \frac{1}{2}.$$

The function A has coalescing eigenvalues only at the origin, which is a nongeneric coalescing point; the eigenvalues of B coalesce along the circle of radius $\frac{1}{2}$ centered at the origin in the xy-plane. The search is performed on the cube $\Omega = [-1, 1]^3$, with `tolls` $= 10^{-1}$ and `tollp` $= \frac{\pi}{6}$.

In both cases the algorithm returned $\alpha_1(1) = \alpha_2(1) = 0$; that is, no coalescing point is detected, and integration proceeded always with the maximum allowed step size.

Interestingly, however, when we use x^3 instead of x^2 in the definition of the function A, then the code successfully finds the coalescing point at the origin. This fact suggests that the algorithm is capable of locating coalescing points of "odd multiplicity," but the theory for this case has not yet been developed (see [6] for the real valued case depending on two parameters).

Example 4.3. This example is motivated by work in (computational) mathematical physics of Wilkinson, Walker, and coworkers; see [19, 17, 18]. The issue is to study the spatial distribution of coalescing points (they are called degeneracies in the cited works) for parameter dependent random matrix models. We refer to the above cited works for the physical relevance of this study; here, we report on the performance of our algorithm for the model in question.

So, we consider the following function $A \in \mathcal{C}^\omega([0,2\pi]^3, \mathbb{C}^{n\times n})$, which is periodic in each component of x:

$$A(x) \;=\; \sum_{j=1}^{3} \big[A_{2j-1}\cos(x_j) + A_{2j}\sin(x_j)\big] \;, \quad x \in [-\pi,\pi]^3 \;, \tag{4.1}$$

where $A_1, \dots, A_6$, are random matrices selected by independent samples from the *Gaussian unitary ensemble* (GUE for short). Since the GUE is invariant under unitary transformations, and we want A to be Hermitian, each A_j should be a Hermitian matrix whose entries are independently distributed elements from the Gaussian distribution with mean 0 and variance 1. Hence, we will select $A_j = B_j + iC_j$, with $B_j = B_j^T$ and $C_j = -C_j^T$ with entries being independently Gaussian distributed. (In practical terms, we generated the upper triangular part of all these symmetric and antisymmetric matrices with the `randn` command in MATLAB and then imposed symmetry, respectively, antisymmetry.)

Our goal is to find the spatial distribution, i.e. the density, of generic coalescing points for the function A of (4.1), as a function of the dimension n. To do this, we will effectively attempt to *count* the number of coalescing points in the cube $[-\pi,\pi]^3$; call $M(n)$ the number of degeneracies in this cube. The outstanding difficulty in doing this computation is that there are a lot of degeneracies, and the computation becomes quite expensive as n grows. We notice that—because of periodicity in x—the number of degeneracies inside any cube in $\mathbb{R}^3$ of edge 2π will be the same; we further notice that the total number of degeneracies inside the cube $[-\pi,\pi]^3$ will be twice the number of degeneracies inside the half-cube $[-\pi,\pi]^2 \times [0,\pi]$, a fact which we used to reduce the overall computational task.

Some theoretical results on the expected value of $M(n)$ are available in the limit as $n \to \infty$, and will be checked against our computations. Indeed, in [17, 18] it is shown that the density of degeneracies is given by

$$\frac{2\sqrt{\pi}}{3}(\rho\sigma)^3 \;,\; \sigma = \sqrt{3} \;,$$

where ρ is the so-called local density of states. Further, using Wigner's semicircle law on the density of states (that is, on the distribution of the eigenvalues of random Hermitian matrices), Walker and Wilkinson in [18] arrived at the following remarkable formula:

$$M(n) \overset{\text{asy}}{\sim} \frac{512 n^{5/2}}{135\sqrt{3\pi}} \;, \tag{4.2}$$

whose most noteworthy feature, of course, is the term $n^{5/2}$. Finally, Walker and Wilkinson in [18] report on results of some numerical experiments they performed on

TABLE 2

Average number of computed degeneracies, M, and standard deviation "std," over 5 *ensembles of size n. Subdivision with edges of length $2\pi/N$. The column "% Error" refers to the error with respect to the formula* (4.2).

n	N	M	std	% Error
5	10	55	6	20%
	20	57	5	18%
	40	57	5	17%
10	20	381	19	2.4%
	40	386	22	1.2%
	80	391	23	0.4%
20	30	2148	15	2.7%
	60	2192	16	0.83%
	120	2203	18	0.3%
30	40	5920	85	2.8%
	80	6019	88	1.2 %
40	50	12113	78	3.1 %
	100	12290	79	1.7 %
50	60	21461	229	1.7%
	120	21747	240	0.4%

the model (4.1) to validate (4.2); however, they did not provide any insight into what technique they used to compute the number of degeneracies. We stress that by virtue of our subdivision technique, we will be providing a very reliable estimate on the size of the smallest box containing a coalescing point, and thus a computational validation of the theoretical results of [18].

In Table 2, we report on the average values of $M(n)$, for $n = 5, 10, \dots, 50$, obtained by averaging (to the closest integer) over 5 ensembles, each ensemble being a different sixtuplet of random Hermitian matrices, computed by our algorithm. In the table, after the value of n, we report the value of N: this refers to the subdvision of the initial cube $[-\pi, \pi]^3$ into N^3 cubes each of edge $2\pi/N$ (see section 3.2). The next two columns are the computed (average) value of M, followed by the standard deviation "std," both rounded to the nearest integer, and the last column is the percentage (relative) deviation with respect to the asymptotic formula (4.2). Looking at this last column, it is apparent that (except for the case of $n = 5$, which is hardly in the asymptotic regime of large n) there is a very good agreement between our computational results and (4.2), a fact which is further highlighted in Figure 7 where we show the line of best fit for $M(n)$ (in the least squares sense) obviously corroborating the validity of the $n^{5/2}$ law; to be precise, the least squares fit of a curve cn^p on the data for $n = 10, \dots, 50$, gave us $p = 2.493$ and $c = 1.256$ (cf. with (4.2)).

We now make a few extra observations based on our computations, which suggest some interesting statistical behavior associated with the coalescing points/eigenvalues.

(1) The distribution of coalescing points in the cube is seemingly uniform. To support this claim, we performed (and passed) several χ^2 goodness-of-fit tests (with a 5% significance level) with varying number of bins for the spatial distribution in the three coordinate axes for size $n = 50$.

(2) In Figure 8, we show plots of the distribution of eigenvalues corresponding to coalescing points (on the half-cube) for five random realizations of the model (4.1) with $n = 50$. Next to it, we also show the distribution of the eigenvalues for the sum of three independently and identically distributed Hermitian matrices of dimension 50 for 1000 realizations, to exemplify the distribution of the eigenvalues of random Hermitian matrices according to the semicircle

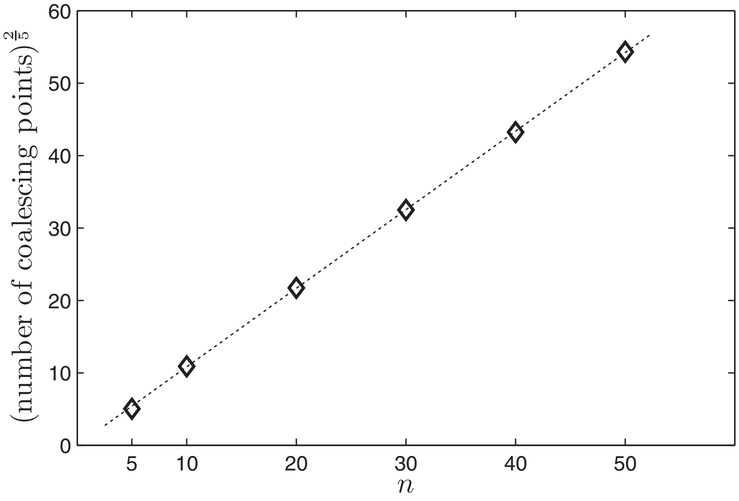

FIG. 7. *Dependence of number of coalescing points on dimension n, Example* 4.3.

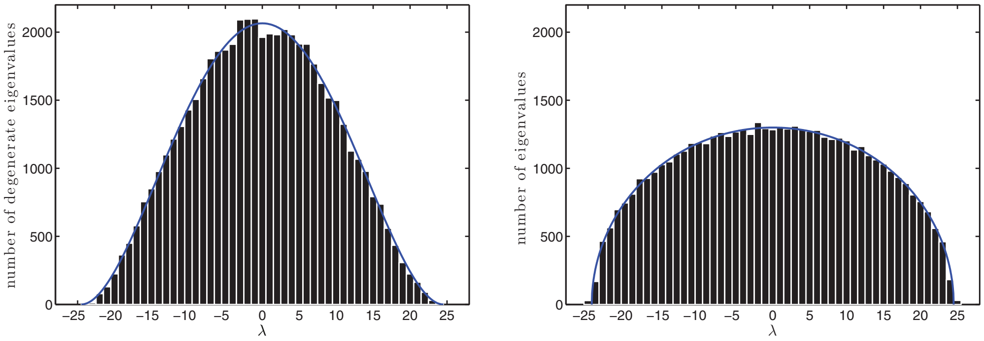

FIG. 8. *Distribution of degeneracies for a realization of* (4.1) *with $n = 50$ (a total of* 54,367 *eigenvalues), and of the eigenvalues of the sum of three random (GUE) Hermitian matrices, $n = 50$, for* 1000 *realizations.*

law, already for such a moderate value of n. Note that with five realizations of the model (4.1) with $n = 50$, we are plotting the distribution of about 50,000 coalescing eigenvalues. Our experimental evidence strongly supports that the coalescing eigenvalues are distributed according to the power law $c(\rho(E))_+^4$, where $\rho(E)$ indicates the density of states (cf. with [19]), and it is given by $\rho(E) = \frac{1}{\pi\sqrt{3}}(n - E^2/12)^{1/2}$. [We are using the notation $z_+ = \max(z, 0)$.] We note that for this density of states we have

$$\int_{-2\sqrt{3n}}^{2\sqrt{3n}} \rho(E)\, dE = n\,.$$

This density of states should be compared with Wigner's semicircle law for the distribution of the eigenvalues of random Hermitian matrices. To witness, on Figure 8 on the left we superimposed the curve $c(\rho(E))_+^p$, where $c \approx 742.47$ and $p \approx 3.9$ were found by nonlinear regression on the ensemble with $n = 50$.

Remark 4.4. We want to emphasize that our algorithm is tested rather severely by this problem. Although our method is designed to find isolated coalescing points when there are few and far between of them in the region of interest, the method is actually very robust, and it is able to handle quite well this special situation in which we have an isolated coalescing point basically in a small neighborhood of any point in the starting cube.

Also our step size selection algorithm is tested very severely by these computations, and the algorithm proved quite robust as well: for example, for $n = 50$ and $N = 120$ (recall that we work on the half-domain $[-\pi, \pi]^2 \times [0, \pi]$) we needed a grand total of about 1.5×10^7 steps along the meridians and 10^9 steps along the parallels, with the number of failed steps being on the order of 0.2% along meridians and of 1% along the parallels.

Remark 4.5. An interesting phenomenon occurs when we considered the model (4.1) with the Hermitian matrices A_{2j-1}, A_{2j} $(j = 1, 2, 3)$, being banded with small bandwidth, rather than full. In this case, we are not aware of any study linking the number of degeneracies to the dimension n, and our experiments show that the relation between the number of degeneracies and the dimension n obeys a different power law, which depends also on the bandwidth. For example, for pentadiagonal matrices, the power law we found is of the type cn^p with $p \approx 2.7$.

Example 4.6. This is a revisitation of Example 4.3. Our algorithm, tracking all of the CIs of the model (4.1), becomes computationally very burdensome already when the size of the matrices reaches a few hundreds. This is not just due to the need to find eigendecompositions of larger matrices, but more importantly to the very large number of continuation steps which we need to take as a result of the large number of conical intersections and the associated effort of the algorithm in following the (very sensitive) eigenvectors involved in the CIs (hence, very small step sizes need to be chosen in order to resolve the CI). To illustrate, consider the ensamble (4.1) with dimension $n = 200$ on the cube $[0, \pi]^3$; for this, we expect nearly 90,000 CIs for each realization.

Here we report on some experiments we have performed on this problem. We consider again five realizations taken from the parametrized ensemble (4.1), with dimension $n = 200$, and the cube $\Omega \equiv [0, \pi]^3$. However, instead of attempting to find all of the CIs, we approximate only those relative to the first and last $d + 1$ eigenvalues, and then extrapolate from this limited data the sought information for the full set of CIs. If successful, this strategy will significantly reduce the required computational effort; for example, the number of CIs that involve the nine outermost pair of eigenvalues is about 1,250. The concern now is how to verify that we can successfully extrapolate from this limited data set, and the primary consideration becomes how to make sure that we are successfully approximating the expected number of CIs that involve any given pair of consecutive eigenvalues of A in Ω. Our approach is based on the following reasoning, which renders remarkably accurate results.

Let $\lambda_1 \geq \cdots \geq \lambda_n$ be the eigenvalues of a function A taken from the parametrized ensemble (4.1). From [17], we know that the (local) density of degeneracies is given by

$$\mathcal{D} = \frac{2\sqrt{\pi}}{3} (\rho(E))^3 ,$$

where $\rho(E)$ is the density of states. For our function A, $\mathcal{D}$ gives the approximate number of CIs (degenerate states) involving a given pair of eigenvalues inside a unit of volume in parameter space, where E is the corresponding value of the degenerate

TABLE 3

d	computed	predicted	% Error
4	277.6	287.7	3.5%
8	982.4	1006.2	2.4%

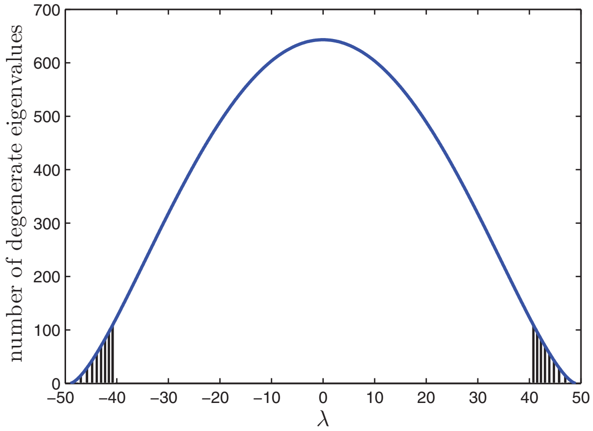

FIG. 9.

state. Our goal is to predict the value of the degenerate state corresponding to each pair of eigenvalues of A.

Let

$$F(E) = \int_E^{2\sqrt{3n}} \rho(s)\, ds\,.$$

For any integer k, with $0 \le k \le n$, let E_k be so that $F(E_k) = k$. By construction, we expect that each interval $[E_{k-1}, E_k]$ contains exactly one eigenvalue of A, for $k = 1, \dots, n$. In other words, λ_k is expected to lie inside $[E_{k-1}, E_k]$ for any $k = 1, \dots, n$. Guided by our numerical experiments, we make the following empirical statement: we expect the kth degenerate state to occur at E_k. Therefore, we expect the number of CIs relative to the largest $d+1$ eigenvalues, inside the cube $[0,\pi]^3$, to be given by

$$c \sum_{k=1}^{d} n_0(E_k)^3 \ , \ c = \pi^3 \frac{2}{3}\sqrt{\pi}\,. \tag{4.3}$$

Based upon the above, we computed CIs relative to the d most extremal pairs of eigenvalues (both largest and smallest). In Table 3, we report on the results of our numerical experiments, averaged over five realizations for (4.1), with dimension $n = 200$. The column "computed" refers to the actual count of CI we found, the column "predicted" is the value from (4.3), and "Error" is the relative error between computed and predicted.

Finally, based upon the data for $d = 8$, we fit by nonlinear regression the curve $cn_0(\lambda)^p$ obtaining $p \approx 2.9963$ and $c \approx 35.838$, both remarkably close to the expected values $p = 3$ and $c = 2\pi^{7/2}/3$. To illustrate, in Figure 9 we show the computed curve of best fit.

Acknowledgment. The help of M. G. Gasparo on a preliminary version of this work is gratefully acknowledged.

REFERENCES

[1] S. Basu, R. Pollack, and M.-F. Roy, *Algorithms in Real Algebraic Geometry*, 2nd ed., Springer-Verlag, New York, 2006.

[2] M. V. Berry, *Quantal phase factors accompanying adiabatic changes*, R. Soc. Lond. Proc. Ser. A Math. Phys Eng. Sci., 392 (1984), pp. 45–57.

[3] 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 (1991), pp. 1–40.

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

[5] L. Dieci, M. G. Gasparo, A. Papini, and A. Pugliese, *Locating coalescing singular values of large two-parameter matrices'*, Math. Comput. Simulation, 81 (2011), pp. 996–1005.

[6] L. Dieci and A. Pugliese, *Two-parameter SVD: Coalescing singular values and periodicity*, SIAM J. Matrix Anal. Appl., 31 (2009), pp. 375–403.

[7] L. Dieci and A. Pugliese, *Singular values of two-parameter matrices: An algorithm to accurately find their intersections*, Math. Comput. Simulation, 79 (2008), pp. 1255–1269.

[8] L. Dieci and A. Pugliese, *Hermitian matrices depending on three parameters: Coalescing eigenvalues*, Linear Algebra Appl., 436 (2012), pp. 4120–4142.

[9] T. Kato, *A Short Introduction to Perturbation Theory for Linear Operators*, Springer-Verlag, New York, 1982.

[10] O. Koch and C. Lubich, *Dynamical low-rank approximation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 434–454.

[11] T. G. Kolda, R. M. Lewis, and V. Torczon, *Optimization by direct search: New perspectives on some classical and modern methods*, SIAM Rev., 45 (2003), pp. 385–482.

[12] A. P. Seyranian and A. A. Mailybaev, *Multiparameter Stability Theory with Mechanical Applications*, World Scientific, Singapore, 2003.

[13] H. D. Simon and H. Zha, *Low-rank matrix approximation using the Lanczos bidiagonalization process with applications*, SIAM J. Sci. Comput., 21 (2000), pp. 2257–2274.

[14] A. Srikantha Phani, J. Woodhouse, and N. A. Fleck, *Wave propagation in two-dimensional periodic lattices*, J. Acoust. Soc. Amer., 119 (2006), pp. 1995–2005.

[15] A. J. Stone, *Spin-orbit coupling and the intersection of potential energy surfaces in polyatomic molecules*, R. Soc. Lond. Proc. Ser. A Math. Phys. Eng. Sci., 351 (1976), pp. 141–150.

[16] J. von Neumann and E. Wigner, *Eigenwerte bei adiabatischen prozessen*, Phys. Z., 30 (1929), pp. 467–470.

[17] P. N. Walker, M. J. Sanchez, and M. Wilkinson, *Singularities in the spectra of random matrices*, J. Math. Phys., 37 (1996), pp. 5019–5032.

[18] P. N. Walker and M. Wilkinson, *Universal fluctuations of Chern integers*, Phys. Rev. Lett., 74 (1995), pp. 4055–4058.

[19] M. Wilkinson and E. J. Austin, *Densities of degeneracies and near-degeneracies*, Phys. Rev. A(3), 47 (1993), pp. 2601–2609.

[20] D. R. Yarkony, *Conical intersections: The new conventional wisdom*, J. Phys. Chem. A, 105 (2001), pp. 6277–6293.