SIAM J. SCI. COMPUT.
Vol. 30, No. 2, pp. 637–656

CONTINUATION OF INVARIANT SUBSPACES IN LARGE BIFURCATION PROBLEMS*

DAVID BINDEL†, JAMES DEMMEL‡, AND MARK FRIEDMAN§

Abstract. We summarize an algorithm for computing a smooth orthonormal basis for an invariant subspace of a parameter-dependent matrix, and describe how to extend it for numerical bifurcation analysis. We adapt the continued subspace to track behavior relevant to bifurcations, and use projection methods to deal with large problems. To test our ideas, we have integrated our code into MATCONT, a program for numerical continuation and bifurcation analysis.

Key words. smooth eigendecompositions, invariant subspaces, continuation, bifurcation analysis, Ritz-Galerkin methods

AMS subject classifications. 65F15, 65F50, 37M20, 65H20

DOI. 10.1137/060654219

1. Introduction.

Parameter-dependent Jacobian matrices provide important information about dynamical systems

$$\frac{du}{dt} = f(u,\alpha), \text{ where } u \in \mathbb{R}^n,\ \alpha \in \mathbb{R},\ f(u,\alpha) \in \mathbb{R}^n. \tag{1.1}$$

For example, to analyze stability at branches $(u(s),\alpha(s))$ of steady states

$$f(u,\alpha) = 0, \tag{1.2}$$

we look at the linearization $A(s) = D_u f(u(s),\alpha(s))$. If the system comes from a spatial discretization of a partial differential equation, then $A(s)$ will typically be large and sparse. In this case, an invariant subspace $\mathcal{R}(s)$ corresponding to a few eigenvalues near the imaginary axis provides information about stability and bifurcations.

Recently, we developed with collaborators the *CIS algorithm* for the continuation of invariant subspaces of a parameter-dependent matrix [17, 20, 26, 27, 7]. In this paper, we extend the CIS algorithm to make it more suitable to numerical bifurcation analysis. Our goal is to extend numerical bifurcation techniques developed for small systems to larger systems. We also wish to ensure that bifurcations are detected reliably; this goal is especially relevant when the linearized system is nonnormal, so that a small perturbation may result in a large change to the eigenvalues [43, 44]. To this end, we make the following contributions to the development of the method: we introduce logic to adapt or reinitialize the subspace during continuation so that it is always well defined and always includes the least-stable modes that are relevant to bifurcation analysis; we extend the algorithm to use Galerkin projection methods when n is large and direct methods are expensive; and we integrate our method

*Received by the editors March 13, 2006; accepted for publication (in revised form) June 16, 2007; published electronically February 14, 2008.
http://www.siam.org/journals/sisc/30-2/65421.html
†Department of Mathematics, Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (dbindel@cims.nyu.edu).
‡Computer Science Division and Department of Mathematics, University of California, Berkeley, CA 94720 (demmel@cs.berkeley.edu).
§Mathematical Sciences Department, University of Alabama, Huntsville, AL 35899 (friedman@math.uah.edu). This author was supported under NSF DMS-0209536 and NSF ATM-0417774.

into CL_MATCONT [18], a bifurcation analysis package in MATLAB. Some initial results in this direction are reported in [8] and [25].

The CIS algorithm consists of a predictor based on first derivative information, and a corrector based on iterative refinement of an approximate invariant subspace (see [41, 16] and the references therein). The algorithm evaluates a smoothly varying orthonormal basis for $\mathcal{R}(s)$ at sample points $s_0 < s_1 < \cdots < s_{N-1} < s_N$. This basis approximately minimizes arclength over all orthonormal bases for $\mathcal{R}(s)$, in a sense we will make precise in section 2.1. The step size is adapted so that $h_i = s_i - s_{i-1}$ decreases when $\mathcal{R}(s)$ changes fast and increases when $\mathcal{R}(s)$ changes slowly. When the eigenvalues corresponding to $\mathcal{R}(s)$ come too near the rest of the spectrum, the continuation procedure breaks down. In this case, the size of the continued subspace is adapted, and continuation proceeds with a larger or smaller subspace.

The rest of the paper is organized as follows. After discussing related work in the remainder of this section, we briefly describe results on existence and uniqueness of continuously-defined invariant subspaces. In section 3, we describe the CIS algorithm and our new algorithms for initializing and updating the invariant subspace during the continuation process. In section 4, we describe how to modify the CIS algorithm to use projection methods. In section 5, we illustrate the usefulness of the modified algorithm in bifurcation analysis through the solution of a model problem in CL_MATCONT. We conclude and present our plans for future work in section 6.

1.1. Related work. The local behavior of eigendecompositions and other matrix factorizations when viewed as matrix functions is of long-standing interest, and is treated in detail in the book by Stewart and Sun [42], or in the standard text by Kato [35]. The local behavior of invariant subspaces can be analyzed by representing the subspaces in terms of their orthogonal departure from a reference space; such analysis leads directly to an algebraic Riccati equation. In [16], this Riccati equation was used as the basis for a unified analysis of several algorithms for refining approximate invariant subspaces; and in more recent work [10], new algorithms for invariant subspace approximation are proposed which combine a Galerkin approximate solution to an algebraic Riccati equation with the subspace construction ideas of the Jacobi–Davidson algorithm. In [24], Edelman and his colleagues proposed a more global approach to the analysis of linear algebra algorithms based on Grassmann manifolds and Stiefel manifolds (manifolds of subspaces and of orthonormal subspace bases, respectively); this approach has inspired several new methods for invariant subspace refinement, four of which are summarized and analyzed in [1].

No algorithm can produce globally continuous eigendecompositions, even for the set of diagonalizable matrices. However, one can smoothly define an invariant subspace basis along a path through matrix space, assuming the path crosses no singularities that would render the subspace discontinuous. In [19], a variety of continuous eigendecompositions for one-parameter matrix functions are described, including continuous Schur and block Schur decompositions. In a paper by Govaerts, Guckenheimer, and Khibnik [33], which motivated our work on invariant subspace continuation, a low-dimensional invariant subspace of the Jacobian matrix, corresponding to the eigenvalues with largest real parts, was computed at each point along a continuation path and used to detect Hopf bifurcations via the bialternate matrix product. The authors concluded that subspace reduction can be combined with complicated bifurcation computations and should be tried for large problems.

The CIS algorithm was presented and analyzed in [17] and further studied in [20, 26], with additional practical developments in [27] and [7]. The algorithm of

[20] constructs a smooth block 2-by-2 Schur decomposition; in [21], the approach is extended to the case of more blocks, and a new method is proposed to compute a smooth similarity reduction to block bidiagonal form. In [22], the approach described in [17] for using subspace continuation to compute connecting orbits between equilibria was extended to compute connecting orbits between periodic orbits. To continue low-dimensional invariant subspaces of sparse matrices, the authors of [6] used a bordered Bartels–Stewart algorithm to solve each corrector iteration; in [9], this approach is combined with ideas from [20, 26]. Though [6] and [9] deal with methods for sparse matrices, they differ from our current work in that they use different predictors and correctors, and they do not analyze and update the subspace during continuation to ensure it retains all information relevant to bifurcations.

Numerical continuation for large nonlinear systems arising from ODEs and discretized PDEs is an active area of research, and the idea of subspace projection is common in many methods being developed. The continuation algorithms are typically based on Krylov subspaces, or on recursive projection methods which use a time integrator instead of a Jacobian multiplication as a black box to identify the low-dimensional invariant subspace where interesting dynamics take place; see, e.g., [3, 40, 31, 12, 32, 23, 28, 11, 14].

2. Continuous invariant subspaces. Let $A \in C^k([0,1],\ \mathbb{R}^{n\times n})$ be a k-times continuously differentiable parameter-dependent matrix. We can write the spectrum $\Lambda(s)$ of $A(s)$ as n continuous functions $\lambda_1(s),\ldots,\lambda_n(s)$ [35]. At parameter values where $\lambda_i(s)$ is a multiple eigenvalue, $\lambda_i(s)$ may not be differentiable, and it may be impossible to define a continuous right eigenvector. However, $\lambda_i(s)$ is a C^k function with a C^k right eigenvector as long as $\lambda_i(s)$ has algebraic multiplicity 1. More generally, define

$$\begin{aligned} \Lambda_1(s) &:= \{\lambda_i(s)\}_{i=1}^{m}, \\ \Lambda_2(s) &:= \{\lambda_i(s)\}_{i=m+1}^{n}, \\ \Lambda(s) &:= \Lambda_1(s) \cup \Lambda_2(s). \end{aligned} \tag{2.1}$$

While $\Lambda_1(s)$ and $\Lambda_2(s)$ remain disjoint, there is a well-defined maximal right invariant subspace $\mathcal{R}(s)$ corresponding to $\Lambda_1(s)$, and $\mathcal{R}(s)$ is C^k. We review this fact from the perspective of differential equations in section 2.2, and from a more algebraic perspective in section 2.3.

In what follows, we will primarily use the Frobenius matrix norm: $\|A\|_F = \sqrt{\mathrm{tr}(A^TA)}$. We also assume that complex conjugate pairs are not split between Λ_1 and Λ_2.

2.1. The geometry of subspaces. We begin with a brief review of the geometry of subspaces and orthonormal bases (see [24] for a more complete treatment). The *Stiefel manifold* $\mathrm{Stief}(n,m)$ is the set of matrices with orthonormal columns:

$$\mathrm{Stief}(n,m) := \{Z \in \mathbb{R}^{n\times m} : Z^TZ = I\}, \tag{2.2}$$

where $m \le n$. We can also write

$$\mathrm{Stief}(n,m) = \{QI_{n,m} : Q \in O(n), I_{n,m} = \text{leading } m \text{ columns of } I_n\}. \tag{2.3}$$

Well-known examples of Stiefel manifolds are the unit sphere (for $m = 1$) and the orthogonal group $O(n)$ (for $m = n$).

The *Grassmann manifold* $\mathrm{Grass}(n,m)$ is the set of all m-dimensional subspaces of $\mathbb{R}^n$. We represent elements of $\mathrm{Grass}(n,m)$ by equivalence classes of members of $\mathrm{Stief}(n,m)$ spanning the same space. That is,

$$\mathrm{Grass}(n,m) = \mathrm{Stief}(n,m)/\left[Z \sim ZU,\, U \in O(m)\right]. \tag{2.4}$$

The tangent space at $Z_0 \in \mathrm{Stief}(n,m)$ is a direct sum of two orthogonal spaces: the vertical space and the horizontal space. The *vertical space* is

$$\{\Delta Z \in \mathbb{R}^{n\times m} : \Delta Z = Z_0 H_{11} \text{ and } H_{11} \in \mathbb{R}^{m\times m} \text{ is skew}\}, \tag{2.5}$$

and the *horizontal space* is

$$\{\Delta Z \in \mathbb{R}^{n\times m} : Z_0^T\, \Delta Z = 0\}. \tag{2.6}$$

The set of matrices in $\mathrm{Stief}(n,m)$ spanning the same space as Z_0 is $\{Z \in \mathrm{Stief}(n,m) : Z = Z_0 U,\ U \in O(m)\}$. The vertical directions are exactly the tangents to this set. So vertical motion "spins" vectors without changing the subspace, while horizontal motion changes the subspace spanned.

We define the differentiable structure of $\mathrm{Grass}(n,m)$ in terms of the structure of $\mathrm{Stief}(n,m)$: a path $\mathcal{Z}(s)$ in $\mathrm{Grass}(n,m)$ is C^k if there is a C^k basis $Z : [0,1] \to \mathrm{Stief}(n,m)$ such that $\mathcal{Z}(s) = \mathrm{span}(Z(s))$. This basis is not unique; however, given a basis $Z_0 \in \mathrm{Stief}(n,m)$ for $\mathcal{Z}(0)$, there is a unique C^k basis starting from Z_0 which moves only horizontally. We describe the basis in the following lemma.

LEMMA 2.1. *Let $\mathcal{Z} : [0,1] \to \mathrm{Grass}(n,m)$ be a C^k parameter-dependent space $(k > 0)$. Then for any $Z_0 \in \mathrm{Stief}(n,m)$ such that $\mathcal{Z}(0) = \mathrm{span}(Z_0)$, there is a unique C^k basis $Z : [0,1] \to \mathrm{Stief}(n,m)$ for $\mathcal{Z}(s)$ such that $Z(0) = Z_0$ and*

$$Z(s)^T Z'(s) = 0. \tag{2.7}$$

This basis minimizes the Euclidean arclength

$$l(Z) = \int_0^1 \|Z'(s)\|_F\, ds \tag{2.8}$$

over all C^k orthonormal bases for $\mathcal{Z}(s)$.

Proof. Let $\hat{Z} : [0,1] \to \mathrm{Stief}(n,m)$ be one C^k orthonormal basis for $\mathcal{Z}$. Any other C^k orthonormal basis for $\mathcal{Z}$ can be written $Z = \hat{Z}U$ for some C^k function $U : [0,1] \to O(m)$. By the Pythagorean theorem,

$$\|(\hat{Z}U)'\|_F^2 = \|(I - \hat{Z}\hat{Z}^T)(\hat{Z}U)'\|_F^2 + \|\hat{Z}\hat{Z}^T(\hat{Z}U)'\|_F^2, \tag{2.9}$$

where the first term corresponds to horizontal motion, and the second term to vertical motion. Since the Frobenius norm is invariant under unitary transformations, we can show that the first term depends only on $\mathcal{Z}$, and not on the particular choice of basis:

$$\|(I - \hat{Z}\hat{Z}^T)(\hat{Z}U)'\|_F = \|(I - \hat{Z}\hat{Z}^T)(\hat{Z}'U + \hat{Z}U')\|_F \tag{2.10}$$

$$= \|(I - \hat{Z}\hat{Z}^T)\hat{Z}'U\|_F \tag{2.11}$$

$$= \|(I - \hat{Z}\hat{Z}^T)\hat{Z}'\|_F. \tag{2.12}$$

By again using unitary invariance of the norm, we rewrite the second term as

$$\|\hat{Z}\hat{Z}^T(\hat{Z}U)'\|_F = \|\hat{Z}^T(\hat{Z}U)'\|_F = \|(\hat{Z}U)^T(\hat{Z}U)'\|_F. \tag{2.13}$$

Therefore, the minimum attainable arclength should occur when

$$0 = (\hat{Z}U)^T(\hat{Z}U)' = U^T(\hat{Z}^T\hat{Z}'U + U') \tag{2.14}$$

or, equivalently,

$$U' = -\hat{Z}^T\hat{Z}'U. \tag{2.15}$$

By the standard theory for linear ODEs, there is a unique U which satisfies (2.15) together with the initial condition $\hat{Z}(0)U(0) = Z_0$. Therefore, there is a unique orthonormal basis $Z = \hat{Z}U$ which satisfies (2.7) and $Z(0) = Z_0$. Furthermore, Z has minimal arclength. ∎

2.2. Differential equation characterization. We can also prove the existence of a C^k invariant subspace by writing a differential equation for a Schur factorization. This is the approach used in [19, 17, 20]; we summarize their result in the following theorem.

THEOREM 2.2 (see [19, 17, 20]). *Suppose $\Lambda_1(s)$ and $\Lambda_2(s)$ are disjoint for all $s \in [0,1]$. Then there is an orthogonal matrix Q and block upper triangular matrix T, each with C^k dependence on s, so that*

$$A(s) = Q(s)T(s)Q(s)^T \tag{2.16}$$

$$= \begin{bmatrix} Q_1(s) & Q_2(s) \end{bmatrix} \begin{bmatrix} T_{11}(s) & T_{12}(s) \\ 0 & T_{22}(s) \end{bmatrix} \begin{bmatrix} Q_1(s) & Q_2(s) \end{bmatrix}^T, \tag{2.17}$$

where $Q_1(s) \in \mathbb{R}^{n\times m}$ is a basis for the subspace $\mathcal{R}(s)$ corresponding to $\Lambda_1(s)$, and $Q_2(s) \in \mathbb{R}^{n\times(n-m)}$ is a basis for $\mathcal{R}(s)^\perp$.

2.3. Algebraic characterization. Suppose $A \in C^k([0,1], \mathbb{R}^{n\times n})$ and at some $s_0 \in [0,1]$, $\Lambda_1(s_0)$ and $\Lambda_2(s_0)$ are disjoint. Then by the results in previous sections, there is a (nonunique) continuous block Schur decomposition for s near s_0, which at s_0 is

$$A(s_0) = \begin{bmatrix} Q_1(s_0) & Q_2(s_0) \end{bmatrix} \begin{bmatrix} T_{11}(s_0) & T_{12}(s_0) \\ 0 & T_{22}(s_0) \end{bmatrix} \begin{bmatrix} Q_1(s_0) & Q_2(s_0) \end{bmatrix}^T, \tag{2.18}$$

where the spectrum of $T_{ii}(s_0)$ is $\Lambda_i(s_0)$. Sufficiently near s_0, continuity demands that no nonzero vector in $\mathcal{R}(s)$ be orthogonal to $\mathcal{R}(s_0)$, so we may write $\mathcal{R}(s) = \operatorname{span}(\bar{Q}_1(s))$, where

$$\bar{Q}(s) = \begin{bmatrix} \bar{Q}_1(s) & \bar{Q}_2(s) \end{bmatrix} = Q(s_0) \begin{bmatrix} I & -Y(s)^T \\ Y(s) & I \end{bmatrix}. \tag{2.19}$$

Here $\bar{Q}_1(s) \in \mathbb{R}^{n\times m}$ is a *nonorthonormal* basis for $\mathcal{R}(s)$, which is normalized so that $Q_1(s_0)^T\bar{Q}_1(s) = I$; and $Y(s) \in \mathbb{R}^{(n-m)\times m}$ represents the part of $\bar{Q}_1(s)$ in the directions orthonormal to $Q_1(s_0)$. This function $Y(s)$ must satisfy an algebraic Riccati equation, which we describe in the following lemma.

LEMMA 2.3 (see [17, 20]). *Let $A \in C^k([0,1], \mathbb{R}^{n\times n})$ have a block Schur decomposition at s_0 as in (2.18), where the diagonal blocks of $T(s_0)$ have disjoint spectra. Define*

$$\widehat{T}(s) = \begin{bmatrix} \widehat{T}_{11}(s) & \widehat{T}_{12}(s) \\ E_{21}(s) & \widehat{T}_{22}(s) \end{bmatrix} := Q(s_0)^T A(s) Q(s_0). \tag{2.20}$$

Then for s near s_0, there is a unique, C^k, smallest solution $Y(s) \in \mathbb{R}^{(n-m)\times m}$ to the Riccati equation

$$F(Y) := \widehat{T}_{22}(s)Y - Y\widehat{T}_{11}(s) + E_{21}(s) - Y\widehat{T}_{12}(s)Y = 0, \tag{2.21}$$

and there is a C^k block Schur decomposition

$$A(s) = Q(s)T(s)Q(s)^T, \tag{2.22}$$

where the orthogonal matrix $Q(s)$ is given by

$$Q(s) = \bar{Q}(s)\left(\bar{Q}(s)^T\bar{Q}(s)\right)^{-1/2}, \tag{2.23}$$

$$\bar{Q}(s) = Q(s_0)\begin{bmatrix} I & -Y(s)^T \\ Y(s) & I \end{bmatrix}. \tag{2.24}$$

This theorem is stated in [17] and [20], and extends results proved by Demmel [16], Stewart [41], and Stewart and Sun [42, section V.2].

3. The CIS algorithm: Direct methods. We now describe the CIS algorithm in the case when we can use direct solvers. Much of this work is described in [17, 20, 26, 27]. Here, we emphasize parts of the computation that we perform differently for the dense case. They are also relevant for the sparse case, with modifications described in section 4.

At the highest level, our algorithm is as follows:

1. Choose an initial invariant subspace $Q_1(s_0)$.
2. Compute a continuation step. This involves a predictor-corrector procedure to compute a new subspace basis $\bar{Q}_1(s_1)$ satisfying the linearized constraint $Q_1(s_0)^T\bar{Q}_1(s_1) = I$, or, equivalently, finding the solution $Y(s_1)$ to the Riccati equation described in section 2.3.
3. Normalize the solution. That is, obtain an orthonormal basis $Q_1(s_1)$ that spans the same space as $\bar{Q}_1(s_1)$, and that approximately minimizes the Frobenius arclength as described in section 2.1.
4. Adapt the space and step size to improve convergence and resolve interesting features.

We can continue either $Q_1(s)$ and $T_{11}(s)$ or the full $Q(s)$ and $T(s)$ matrices. Currently, our dense code computes the full Schur factors at each step. When we continue only the first part of the decomposition, as we do in the sparse case, we also compute a few extra eigenvalues from $\Lambda_2(s)$. We use these eigenvalues to decide whether the algorithm should be reinitialized with a different partitioning of the spectrum.

3.1. Initialization. To initialize the algorithm at s_0, we compute a Schur decomposition of $A(s_0)$ and use standard LAPACK routines [2] to sort the decomposition so selected eigenvalues appear in $T_{11}(s_0)$. For bifurcation problems, we assume that only a small part of the spectrum is unstable; therefore, we include in our m-dimensional subspace vectors corresponding to all the unstable eigenvalues as well as a few stable eigenvalues nearest the imaginary axis (see Figure 3.1).

We require that $\Lambda_1(s_0)$ contains any unstable eigenvalues and some specified number of stable eigenvalues; but we may include additional eigenvalues in order to simplify the subsequent continuation process. For example, we include an extra eigenvalue in order to avoid splitting a complex conjugate pair of eigenvalues between $\Lambda_1(s_0)$ and $\Lambda_2(s_0)$. More generally, we would like to choose $\Lambda_1(s_0)$ so that the gap

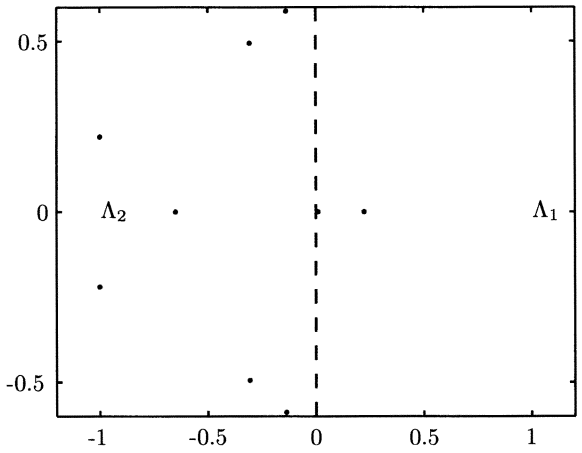

FIG. 3.1. *Selected eigenvalues during initialization.*

between the real parts of the leftmost eigenvalue in $\Lambda_1(s_0)$ and the rightmost eigenvalue in $\Lambda_2(s_0)$ are greater than some threshold. In this way, we hope to keep track of all eigenvalues that might cross the imaginary axis.

In the dense case, the same LAPACK routine used to sort the Schur form also estimates the sensitivity of the selected subspace, and so we may choose a larger subspace if the smallest feasible subspace is very sensitive. Though the cost of the computations at a single point increases as we increase the size of our subspace, continuing a less sensitive subspace will allow us to take larger steps.

We summarize the initialization procedure in Algorithm 1, which is a modification of the original CIS algorithm.

Algorithm 1 CHOOSE AN INITIAL SUBSPACE.

Input: $A(s_0)$,
$n_{\min}$, $n_{\max}$, {bounds on subspace size}
$n_{\text{stableref}}$, {number of stable reference eigenvalues}
ϵ_{gap}, {minimum gap between $\Lambda_1(s_0)$ and $\Lambda_2(s_0)$}

Output: $Q_1(s_0)$ and $T_{11}(s_0)$

Compute a Schur decomposition $A(s_0) = QTQ^T$
$t :=$ real parts of converged eigenvalues sorted in descending order

Find smallest m so that $\begin{cases} n_{\min} \le m \le n_{\max} \\ m \ge (\#\text{ unstable eigenvalues}) + n_{\text{stableref}} \\ t(m) - t(m+1) > \epsilon_{\text{gap}} \end{cases}$

if no such m exists **then**
  error "Spectrum too tightly clustered"
else
  Sort subspace for rightmost m eigenvalues to the front of Q, T
  Return $Q_1 = Q(:, 1:m)$, $T_{11} = T(1:m, 1:m)$
end if

We continue the subspace using an approximate Euler predictor and a Newton corrector, as described below. We use the convergence of the corrector to govern our

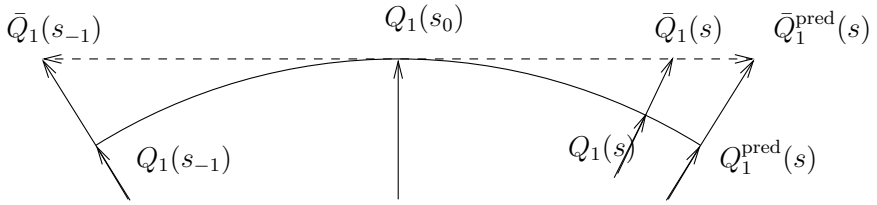

FIG. 3.2. *Choosing a consistent normalization for secant prediction.*

step size: if it converges slowly or fails to converge, then we reduce the step size, or reinitialize the continuation process with a larger or smaller subspace. If the corrector converges quickly, then we increase the step size.

3.1.1. Subspace predictors. We build an Euler predictor (already described before) for $\mathcal{R}(s_1)$ by differentiating the Schur factorization and substituting finite difference approximations for Q' and T'. Alternatively, we could differentiate the Riccati equation (2.21) and substitute a finite difference approximation for Y'. Either way, this gives us

$$T_{22}(s_0)Y_0(s_1) - Y_0(s_1)T_{11}(s_0) = -(s_1 - s_0)E'_{21}(s_1). \tag{3.1}$$

If the derivatives of A are unavailable, then we can also use the trivial predictor $Y_0(s_1) = 0$.

We can also build predictors based on polynomial interpolation, the simplest of which is a secant predictor. To do this, we must consider how consecutive steps are normalized. In a single predictor-corrector step, we normalize the basis for a space $\mathcal{X}$ by requiring that $Q(s_0)^T X = I$; however, this normalization changes with each step. If $\mathcal{R}(s_{-1})$ is the invariant subspace from a previous continuation step, then we must choose a basis $\bar{Q}_1(s_{-1})$ for $\mathcal{R}(s_{-1})$ which is consistent with the current normalization (see Figure 3.2). Because $\bar{Q}_1(s_{-1})$ spans the same space as $Q_1(s_{-1})$, there must be some invertible $B(s_{-1}) \in \mathbb{R}^{m\times m}$ such that

$$\bar{Q}_1(s_{-1}) = Q_1(s_{-1})B(s_{-1}), \tag{3.2}$$

and the normalizing condition is

$$I = Q_1(s_0)^T \bar{Q}_1(s_{-1}) = Q_1(s_0)^T Q_1(s_{-1})B(s_{-1}). \tag{3.3}$$

Therefore

$$B(s_{-1}) = \left(Q_1(s_0)^T Q_1(s_{-1})\right)^{-1}, \tag{3.4}$$

$$\bar{Q}_1(s_{-1}) = Q_1(s_{-1})\left(Q_1(s_0)^T Q_1(s_{-1})\right)^{-1}. \tag{3.5}$$

By linear extrapolation, the secant predictor for $\bar{Q}_1(s_1)$ is

$$\bar{Q}_1^{\text{pred}}(s_1) = Q_1(s_0) + \frac{s_1 - s_0}{s_0 - s_{-1}}\left(Q_1(s_0) - \bar{Q}_1(s_{-1})\right). \tag{3.6}$$

The Riccati unknown has the form $Y(s) = Q_2(s_0)^T \bar{Q}_1(s)$ with $Y(s_0) = 0$, so we can rewrite the predictor (3.6) as

$$Y_0(s_1) = -\frac{s_1 - s_0}{s_0 - s_{-1}} Y(s_{-1}), \tag{3.7}$$

where

$$Y(s_{-1}) = Q_2(s_0)^T \bar{Q}_1(s_{-1}). \tag{3.8}$$

We similarly write higher-order polynomial predictors by choosing a consistent normalization for several steps and using polynomial extrapolation.

3.1.2. Direct Newton corrector iterations. One way to find $\bar{Q}_1(s)$ is to simultaneously solve residual equations for the eigensystem and the normalization:

$$R = \begin{bmatrix} A(s)\bar{Q}_1(s_1) - \bar{Q}_1(s_1)\bar{T}_{11}(s_1) \\ Q_1(s_0)^T \bar{Q}_1(s_1) - I \end{bmatrix} = 0. \tag{3.9}$$

We can compute a Newton step for (3.9) using a bordered Bartels–Stewart algorithm [6]. Alternately, we can eliminate $\bar{T}_{11}(s_1)$ and perform Newton iteration on the Riccati equation (2.21). A Newton step for the Riccati equation can be solved using an ordinary Bartels–Stewart algorithm [29, p. 367].

Newton iterations on the reduced and unreduced systems are equivalent in exact arithmetic, assuming that the initial iterate in the unreduced case satisfies the normalization condition $Q_1(s_0)^T \bar{Q}_1^{\text{pred}}(s_1) = I$. However, while reducing (3.9) to a Riccati equation reduces the problem size by a modest amount, the reduced system will usually be dense, even if (3.9) is sparse. For small problems, we use dense methods, and the loss of sparsity matters little; for large problems, we sidestep the issue by using projection methods, as described in section 4. For medium-sized problems, it may be better to use sparse direct solvers to take Newton steps on the unreduced system of equations, as is done in [6] and in [9].

3.2. Normalizing the solution. Ideally, we would like to choose $Q_1(s)$ to be the unique C^k orthonormal basis for $\mathcal{R}(s)$ with minimal Frobenius arclength, as described in section 2.1. That is, over N steps we would like to minimize $l(Q_1)$, where

$$l(Q_1) := \int_{s_0}^{s_N} \|Q_1'(s)\|_F \, ds. \tag{3.10}$$

Because we cannot compute $l(Q_1)$ exactly, we instead minimize

$$\hat{l}(Q_1) = \sum_{j=1}^{N} \|Q_1(s_j) - Q_1(s_{j-1})\|_F, \tag{3.11}$$

which is a first-order accurate approximation to $l(Q_1)$. In order to minimize $\hat{l}(Q_1)$, we must normalize $Q_1(s_j)$ to minimize $\|Q_1(s_j) - Q_1(s_{j-1})\|_F$ for each j. We perform this normalization via an SVD, as described in the following lemma.

LEMMA 3.1. *Let $\bar{Q}_1(s_1)$ be a basis for $\mathcal{R}(s_1)$ with $Q_1(s_0)^T \bar{Q}_1(s_1) = I$. Let $\bar{Q}_1(s_1) = U\Sigma V^T$ be a singular value decomposition with $U \in \mathbb{R}^{n\times m}$ and $\Sigma, V \in \mathbb{R}^{m\times m}$. Then the orthonormal basis $Q_1(s_1) \in \text{Stief}(n,m)$ for $\mathcal{R}(s_1)$ which minimizes $\|Q_1(s_1) - Q_1(s_0)\|_F$ can be written as*

$$Q_1(s_1) = UV^T. \tag{3.12}$$

Proof. Any orthonormal basis for $\mathcal{R}(s_1)$ can be written UW for some orthogonal W. Finding W to minimize $\|Q_1(s_0) - UW\|_F$ is an orthogonal Procrustes problem [34, 29], and the solution is the polar factor of $U^T Q_1(s_0)$. But $U^T Q_1(s_0) = V^T \Sigma^{-1}$ by the normalization condition on $\bar{Q}_1(s_1)$, so $W = V^T$ is the relevant polar factor. ∎

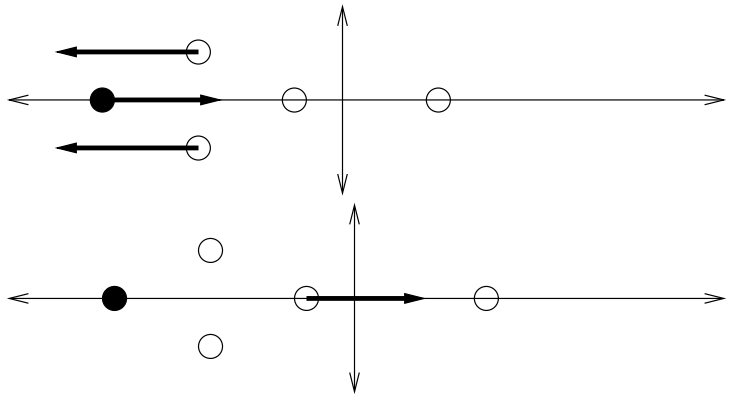

FIG. 3.3. *Examples of overlap and bifurcation. In the top example (overlap), one of the eigenvalues from $\Lambda_1(s)$ (open circles) changes position with one of the eigenvalues of $\Lambda_2(s)$. In the bottom example, an eigenvalue crosses over the imaginary axis (a bifurcation), so that $\Lambda_1(s)$ contains fewer stable eigenvalues.*

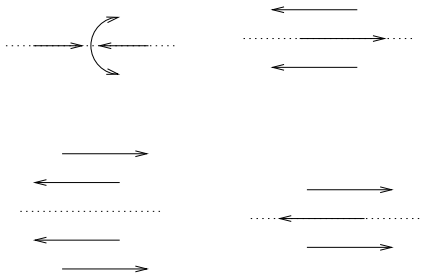

FIG. 3.4. *Generic overlap situations. On the left, two real eigenvalues collide and produce a complex pair (top), and the real parts of two complex conjugate eigenvalue pairs change order (bottom). On the right, a complex conjugate pair and a real eigenvalue change places in two ways.*

3.3. Bifurcations and overlaps. When the CIS algorithm is initialized, the set $\Lambda_1(s_0)$ contains all the unstable eigenvalues of $A(s_0)$ and a few of the stable eigenvalues nearest the imaginary axis. The set $\Lambda_2(s_0)$ lies strictly left of $\Lambda_1(s_0)$ in the complex plane. During continuation, eigenvalues from $\Lambda_1(s)$ may cross the imaginary axis (a bifurcation), or $\Lambda_2(s)$ may cease to lie strictly to the left of $\Lambda_1(s)$ (an overlap). These situations are illustrated in Figure 3.3. When bifurcation or overlap occurs, we reinitialize the continuation procedure.

A *generic* overlap or bifurcation is one which persists when the function $A(s)$ is perturbed. For steady-state continuation problems, the only generic bifurcations are fold bifurcations, in which an isolated real eigenvalue crosses the imaginary axis; and Hopf bifurcations, in which an isolated complex conjugate pair of eigenvalues crosses the imaginary axis. There are four generic types of overlap (see Figure 3.4). In three cases, a single real eigenvalue or complex conjugate pair from $\Lambda_2(s)$ moves right of some element of $\Lambda_1(s)$. In the fourth case, a single eigenvalue from $\Lambda_2(s)$ collides with an eigenvalue from $\Lambda_1(s)$ to form a complex conjugate pair. $Q_1(s)$ corresponding to $\Lambda_1(s)$ will cease to be continuously defined, and we expect that the Newton iteration will not converge. Complex conjugate eigenvalues in the spectrum may also generically collide and become real eigenvalues, but because we do not allow complex conjugate

pairs to be split between $\Lambda_1(s)$ and $\Lambda_2(s)$, this behavior does not result in an overlap.

3.4. Step size and subspace adaptation. Standard bifurcation analysis algorithms [32] involve computing functions of $A(s)$. We adapt these methods to large problems by computing the same functions of the much smaller $T_{11}(s)$. Therefore, we try to ensure that only eigenvalues from $\Lambda_1(s)$ can cross the imaginary axis, so that $T_{11}(s)$ will provide all the relevant information about bifurcations. To prevent eigenvalues from $\Lambda_2(s)$ from crossing the imaginary axis, we adapt the step size and the size of $\Lambda_1(s)$ so that overlaps and bifurcations are not allowed in the same step. We summarize the step size and subspace adaptation logic in Algorithm 2.

When an overlap occurs because two real eigenvalues collide to form a conjugate pair, the Newton iteration will fail to converge. To detect other types of overlap at s, we compute the overlap set:

$$\{(\lambda_i(s), \lambda_j(s)) \in \Lambda_1(s) \times \Lambda_2(s) : \mathrm{Re}(\lambda_i(s)) < \mathrm{Re}(\lambda_j(s))\}.$$

If this set is nonempty, then an overlap has occurred. To decide whether multiple overlaps have occurred, we count the number of $(\lambda_i(s), \lambda_j(s))$ pairs in the overlap set. To avoid double-counting overlaps involving complex eigenvalues, we only count the pairs such that $\mathrm{Im}(\lambda_i(s)) \leq 0$ and $\mathrm{Im}(\lambda_j(s)) \leq 0$.

Only one overlap is allowed in a step. If we detect multiple overlaps, then we retry with a smaller step size until only one overlap is left. If we reach the minimum step size and still have multiple overlaps when continuing from s_i, then we reinitialize the continuation process at s_i so that the overlap set from the failed step belongs entirely to $\Lambda_1(s_i)$ or entirely to $\Lambda_2(s_i)$.

We detect bifurcations by counting the unstable eigenvalues. If the total number of unstable eigenvalues at s_{i+1} differs from the total number of unstable eigenvalues at s_i, then a bifurcation occurred during the step. If this total number is changed by more than one real eigenvalue or one complex conjugate eigenpair, then we assume that multiple bifurcations have occurred, and we try to resolve them by decreasing the step size. If we cannot resolve the behavior with the minimum step size, then the algorithm fails with a diagnostic message. Unless we fail or a bifurcation and an overlap both occur during the step, we assume that $\Lambda_1(s)$ contains all information about bifurcations.

If an overlap or bifurcation occurs in an accepted step from s_i to s_{i+1}, then we will reinitialize the computation at s_{i+1} before attempting another step. This way, the new spectral sets will not overlap, and the new $\Lambda_1(s_{i+1})$ will include no more or fewer eigenvalues than necessary after a bifurcation.

4. The CIS algorithm: Projection methods. We now turn to the case when the dimension n of $A(s)$ is large and we are interested in a space $\mathcal{R}(s)$ of dimension $m \ll n$. In this case, direct methods are expensive; however, if we can multiply by $A(s)$ quickly, then we can use projection methods.

4.1. Choosing a projection space. In the direct case, we considered two spectral sets: $\Lambda_1(s)$, which contains the unstable eigenvalues and a few of the rightmost stable eigenvalues; and $\Lambda_2(s)$, which contains the remaining eigenvalues. In the projection case, we instead consider three spectral sets: $\Lambda_1(s)$, a set of m elements that contains the unstable eigenvalues and a few of the rightmost stable eigenvalues; $\Lambda_2(s)$, a set of $p - m$ elements that contains a few of the rightmost eigenvalues not in $\Lambda_1(s)$; and $\Lambda_3(s)$, a set of $n - p$ elements that contains the remainder of the spectrum that is not computed. Our basic strategy in the projected CIS algorithm is to build a

Algorithm 2 CONTINUE AND ADAPT INVARIANT SUBSPACE OF $A(s)$.

Input: $A(s)$, {matrix-valued function}
s_0, {starting parameter}
h_{initial}, {starting step size}
$h_{\min}$, $h_{\max}$, {bounds on the step size}
Output: $Q(s)$ and $T(s)$

```
Compute initial point Q(s_0), T(s_0) using Algorithm 1.
s := s_0, h := h_initial
while not done do
  Compute a candidate step and candidate step size ĥ
  Test for bifurcation and overlap
  if subspace did not converge then
    Reinitialize at s using Algorithm 1
    Reset step size to h_initial
  else if multiple overlap, multiple bifurcation, or overlap and bifurcation then
    if h > h_min then
      Decrease h
    else if multiple bifurcation then
      error "Could not resolve nongeneric bifurcation"
    else
      Reinitialize at s using Algorithm 1
    end if
  else
    Record the decomposition and diagnostic information
    s := s + h, h := min(h_max, ĥ)
    if bifurcation or overlap occurred in accepted step then
      Reinitialize at s using Algorithm 1
      h := h_initial
    end if
  end if
end while
```

projection space $\mathcal{V}$ of dimension p, where $m < p \ll n$, so that the restriction of $A(s)$ to $\mathcal{V}$ provides good approximations to $\Lambda_1(s)$ and $\Lambda_2(s)$.

4.2. Initialization. During initialization, we may not know how large $\mathcal{V}$ must be to find all the unstable eigenvalues plus a few stable eigenvalues. Therefore, the projected version of the initialization routine computes partial eigendecompositions in a loop using implicitly restarted Arnoldi iteration. While not enough stable eigenvalues converge or there are no sufficiently large gaps between stable eigenvalues in the converged part spectrum, more eigenvalues are requested. If a suitable subspace cannot be found when a specified maximum number of eigenvalues are requested, then the code exits with a diagnostic message.

4.3. Projected normalization and residual equations. Suppose $V \in \mathbb{R}^{p\times n}$ is an orthonormal basis for a projection space $\mathcal{V}$. Recall the n-by-m residual equation

$$A(s)\bar{Q}_1(s) - \bar{Q}_1(s)\bar{T}_{11}(s) = 0.$$

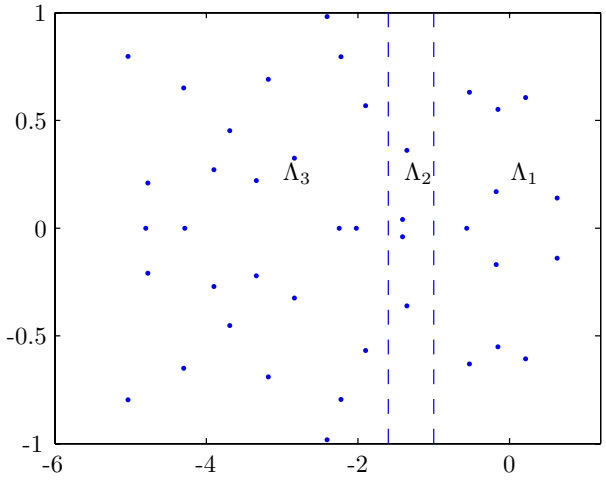

FIG. 4.1. *Eigenvalue sets in the projected CIS algorithm. In practice,* Λ_3 *will contain many more eigenvalues than* Λ_1 *and* Λ_2.

We approximate the equation by assuming that $\bar{Q}_1(s) \approx \bar{Q}_1^h(s) := V\hat{Q}_1(s)$ and choosing $\bar{Q}_1^h(s)$ to satisfy the Galerkin condition

$$0 = V^T\left(A(s)\bar{Q}_1^h(s) - \bar{Q}_1^h(s)\bar{T}_{11}^h(s)\right) \tag{4.1}$$

$$= V^T A(s)V\hat{Q}_1(s) - \hat{Q}_1(s)\bar{T}_{11}^h(s). \tag{4.2}$$

We assume the same normalizing condition we used before:

$$Q_1(s_0)^T\bar{Q}_1^h(s) = \left(V^TQ_1(s_0)\right)^T\hat{Q}_1(s) = I. \tag{4.3}$$

Once $\bar{Q}_1^h(s)$ has been computed, we can use Lemma 3.1 to compute the orthonormal basis $Q_1^h(s)$ for the same space which is closest to $Q_1(s_0)$ in the Frobenius norm. We will let $Q_2^h(s) \in \mathbb{R}^{n\times(p-m)}$ be an orthonormal basis for the orthogonal complement of $\text{span}(Q_1^h(s))$ in $\mathcal{V}$. Though we require continuity of $Q_1^h(s)$, it will not be important for our purposes to continuously define $Q_2^h(s)$.

We typically will use a projection space $\mathcal{V}$ which is itself an approximate maximal invariant subspace computed by an Arnoldi method. Suppose that $A(s_1)\mathcal{V} \subset \mathcal{V}$, and let $V^\perp \in \mathbb{R}^{n\times(n-p)}$ be an orthonormal basis for $\mathcal{V}^\perp$. Then at s_1, solutions to the Galerkin equation (4.2) span invariant subspaces of $A(s_1)$.

If $\mathcal{V}$ is a p-dimensional maximal invariant subspace corresponding to the rightmost part of the spectrum of $A(s_1)$, then we compute the leading 2-by-2 part of a three-by-three block Schur form

$$A(s_1) = \begin{bmatrix} Q_1^h(s_1) & Q_2^h(s_1) & V^\perp \end{bmatrix} \begin{bmatrix} T_{11}^h(s_1) & T_{12}^h(s_1) & T_{13}^h(s_1) \\ 0 & T_{22}^h(s_1) & T_{23}^h(s_1) \\ 0 & 0 & T_{33}^h(s_1) \end{bmatrix} \begin{bmatrix} Q_1^h(s_1) & Q_2^h(s_1) & V^\perp \end{bmatrix}^T.$$

The spectrum of the $T_{11}^h(s)$ block is the continued set of eigenvalues $\Lambda_1(s)$. The $T_{22}^h(s)$ block has a few of the rightmost remaining eigenvalues, which we use to diagnose

overlap. The eigenvalues of the uncomputed block $T_{33}^h(s)$ are part of the spectrum which lies further from the imaginary axis. Figure 4.1 illustrates the three spectral sets corresponding to $T_{11}^h(s)$, $T_{22}^h(s)$, and $T_{33}^h(s)$ in the case when no overlap has occurred.

As in the dense case, we can eliminate $\bar{T}_{11}^h(s)$ from (4.2); we summarize this calculation in the following lemma.

LEMMA 4.1. *Let $V^T Q_1(s_0)$ have the singular value decomposition (SVD)*

$$V^T Q_1(s_0) = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} R^T = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} R^T, \tag{4.4}$$

where $U \in \mathbb{R}^{p\times p}$, $\Sigma \in \mathbb{R}^{m\times m}$, and $R \in \mathbb{R}^{m\times m}$. Let

$$\widehat{T}^h(s) = \begin{bmatrix} \widehat{T}_{11}^h(s) & \widehat{T}_{12}^h(s) \\ E_{21}^h(s) & \widehat{T}_{22}^h(s) \end{bmatrix} := \begin{bmatrix} \Sigma & 0 \\ 0 & I \end{bmatrix} U^T V^T A(s) V U \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & I \end{bmatrix}. \tag{4.5}$$

Then any solution to the Galerkin equation (4.2) *and normalizing condition* (4.3) *can be written as*

$$\hat{Q}_1(s) = U \begin{bmatrix} \Sigma^{-1} \\ \hat{Y}^h(s) \end{bmatrix} R^T, \tag{4.6}$$

where $\hat{Y}^h(s) \in \mathbb{R}^{(p-m)\times m}$ is a solution to the Riccati equation

$$F^h(Y^h(s)) := \widehat{T}_{22}^h(s) Y^h(s) - Y^h(s) \widehat{T}_{11}^h(s) + E_{21}^h(s) - Y^h(s) \widehat{T}_{12}^h(s) Y^h(s) = 0. \tag{4.7}$$

Proof. Let $B(s) = U^T \hat{Q}_1^h(s) R$. By substituting the SVD (4.4) into (4.3), we have

$$I = R \begin{bmatrix} \Sigma & 0 \end{bmatrix} U^T \hat{Q}_1(s) \tag{4.8}$$

$$= R \begin{bmatrix} \Sigma & 0 \end{bmatrix} B(s) R^T. \tag{4.9}$$

If we multiply on the left by R^T and on the right by R, then we have

$$I = \begin{bmatrix} \Sigma & 0 \end{bmatrix} B(s). \tag{4.10}$$

Therefore, for some $Y^h(s) \in \mathbb{R}^{(p-m)\times m}$, $B(s)$ can be written as

$$B(s) = \begin{bmatrix} \Sigma^{-1} \\ Y^h(s) \end{bmatrix} = \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I \\ Y^h(s) \end{bmatrix}. \tag{4.11}$$

Now we substitute $\hat{Q}_1^h(s) = U B(s) R^T$ into the projected residual equation (4.2):

$$V^T A(s) V U \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I \\ Y^h(s) \end{bmatrix} R^T - U \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} I \\ Y^h(s) \end{bmatrix} R^T \bar{T}_{11}^h(s) = 0. \tag{4.12}$$

If we multiply by $\begin{bmatrix} \Sigma & 0 \\ 0 & I \end{bmatrix} U^T$ on the left and by R on the right, then we have

$$\widehat{T}^h(s) \begin{bmatrix} I \\ Y^h(s) \end{bmatrix} = \begin{bmatrix} I \\ Y^h(s) \end{bmatrix} \left(R^T \bar{T}_{11}^h(s) R \right). \tag{4.13}$$

The first row of (4.13) gives an expression for $R^T \bar{T}_{11}^h(s) R$, which we can substitute into the second row to get the Riccati equation (4.7):

$$\begin{aligned} R^T \bar{T}_{11}^h(s) R &= \widehat{T}_{11}^h(s) + \widehat{T}_{12}^h(s) Y^h(s), \\ E_{21}^h(s) + \widehat{T}_{22}^h(s) Y^h(s) &= Y^h(s) \left(R^T \bar{T}_{11}^h(s) R \right) \\ &= Y^h(s) \widehat{T}_{11}^h(s) + Y^h(s) \widehat{T}_{12}^h(s) Y^h(s). \end{aligned}$$

∎

In Theorem 2.3, we saw that for s sufficiently near s_0, the normalized basis for $\mathcal{R}(s)$ corresponded to the minimum norm solution for the Riccati equation (2.21). The norm of the Riccati unknown $Y(s)$ is equal to the distance $\|\bar{Q}_1(s) - Q_1(s_0)\|_F$. We now show that $\|Y^h(s)\|_F$ is similarly related to $\|\bar{Q}_1^h(s) - Q_1(s_0)\|_F$.

LEMMA 4.2. *In the previous lemma, the distance from $\bar{Q}_1^h$ to $Q_1(s_0)$ is*

$$\|\bar{Q}_1^h(s) - Q_1(s_0)\|_F^2 = \|Y^h\|_F^2 + \|\Sigma^{-1}\|_F^2 - m. \tag{4.14}$$

Proof. We decompose $Q_1(s_0)$ and $\bar{Q}_1^h(s)$ into components in three orthogonal spaces spanned by $V^\perp$, VU_1, and VU_2:

$$Q_1(s_0) = V^\perp(V^\perp)^T Q_1(s_0) + VU_1\Sigma R^T, \tag{4.15}$$

$$Q_1^h(s) = VU_1\Sigma^{-1}R^T + VU_2Y^h(s)R^T, \tag{4.16}$$

where the first equation is a consequence of (4.4) and the second equation follows from (4.6). The difference is

$$Q_1(s_0) - Q_1^h(s) = \left\{ \begin{array}{l} V^\perp(V^\perp)^T Q_1(s_0) \\ +VU_1(\Sigma - \Sigma^{-1})R^T \\ +VU_2Y_h(s)R^T \end{array} \right\}. \tag{4.17}$$

Because the three components are orthogonal, the squared Frobenius norm is the sum of the squares of the Frobenius norms; that is

$$\|Q_1(s_0) - Q_1^h(s)\|_F^2 = \left\{ \begin{array}{l} \|V^\perp(V^\perp)^T Q_1(s_0)\|_F^2 \\ +\|VU_1(\Sigma - \Sigma^{-1})R^T\|_F^2 \\ +\|VU_2Y_h(s)R^T\|_F^2 \end{array} \right\}. \tag{4.18}$$

Because multiplication by an orthonormal matrix does not change the Frobenius norm, we can write

$$\|Q_1(s_0) - Q_1^h(s)\|_F^2 = \left\{ \begin{array}{l} \|(V^\perp)^T Q_1(s_0)\|_F^2 \\ +\|\Sigma - \Sigma^{-1}\|_F^2 \\ +\|Y_h(s)\|_F^2 \end{array} \right\} \tag{4.19}$$

$$= \left\{ \begin{array}{l} \|(V^\perp)^T Q_1(s_0)\|_F^2 \\ +(\|\Sigma\|_F^2 + \|\Sigma^{-1}|_F^2 - 2m) \\ +\|Y_h(s)\|_F^2 \end{array} \right\}. \tag{4.20}$$

Note that

$$m = \|Q_1(s_0)\|_F^2 = \|(V^\perp)^T Q_1(s_0)\|_F^2 + \|V^T Q_1(s_0)\|_F^2 \tag{4.21}$$

$$= \|(V^\perp)^T Q_1(s_0)\|_F^2 + \|\Sigma\|_F^2. \tag{4.22}$$

Now substitute

$$\|(V^\perp)^T Q_1(s_0)\|_F^2 = m - \|\Sigma\|_F^2 \tag{4.23}$$

into (4.20) to obtain the desired result. ∎

Therefore, if s_1 is sufficiently near s_0 and $\mathcal{V}$ is itself an invariant subspace of $A(s_1)$ such that $\mathcal{R}(s_1) \subset \mathcal{V}$, then the minimal norm solution to the projected Riccati equation (4.7) corresponds exactly to the minimal norm solution to the Riccati equation (2.21).

4.4. Projected predictors and correctors. The Euler predictor (3.1) is subtly different in the projected case. A projection subspace $\mathcal{V}$ which is an invariant subspace for $A(s_1)$ will generally not contain $\mathcal{R}(s_0)$; consequently, $Q_1(s_0)$ will not correspond to a solution to the projected Riccati equation (4.7) at $s = s_0$. Worse, $E_{21}^h(s_0)$ will usually be nonzero. If we naively differentiate the relation $F^h(Y^h(s)) = 0$ and use the resulting differential equation to form an Euler-like approximation $Y_0^h(s_1)$ starting from a value of 0 for $Y^h(s_0)$, then to first order $F^h(Y_0^h(s_1))$ will be $E_{21}(s_0)$.

We can remedy this problem by requiring $\mathcal{R}(s_0) \subset \mathcal{V}$. However, a more straightforward alternative is to compute a secant prediction $\bar{Q}_1^{\text{pred}}(s_1)$ using (3.6), and then project

$$\bar{Q}_1^{h,\text{pred}}(s_1) = VV^T\bar{Q}_1^{\text{pred}}(s_1). \tag{4.24}$$

The corresponding projected Riccati predictor is then

$$Y_0^h(s_1) = U_2^TV^T\bar{Q}_1^{\text{pred}}(s_1)R. \tag{4.25}$$

In the current code, we use the trivial predictor $Y_0^h(s_1) = 0$, as it has worked well in our test problems.

Once we have a predicted value $Y_0^h(s_1)$, we solve the projected Riccati equation with a Newton iteration, just as we did in the direct methods. We note that the projected matrix $V^TA(s)V$ will usually be dense, and so there seems to be little benefit in solving the unreduced equations. Just as in the direct case, alternate subspace selection methods based on eigenvalues and eigenvectors are possible.

5. Integrating the CIS algorithm into *MATCONT*. In the introduction, we described how invariant subspace continuation can be used to adapt bifurcation analysis methods for small problems in order to analyze much larger systems. In this section, we discuss one example of our work to use the CIS algorithm in this way to extend the bifurcation analysis code MATCONT [18]: using projected test functions to detect and locate Hopf bifurcations.

5.1. Detecting and locating Hopf bifurcations. Let $x(s) = (u(s), \alpha(s)) \in \mathbb{R}^n \times \mathbb{R}$ be a smooth local parameterization of a solution branch of the stationary problem (1.2):

$$f(x(s)) = f(u(s), \alpha(s)) = 0.$$

We write the Jacobian matrix along this path as $A(s) := f_u(x(s))$. A solution point $x(s_0)$ is a *bifurcation point* if $\operatorname{Re}\lambda_i(s_0) = 0$ for at least one eigenvalue $\lambda_i(s_0)$ of $A(s_0)$. The point $x(s_0)$ is a *simple Hopf bifurcation* if the simple eigenvalue $\lambda_i(s_0)$ is a pure imaginary number and $\operatorname{Re}\left(\frac{d\lambda_i}{ds}(s_0)\right) \neq 0$.

A *test function* $\psi(x(s))$ is a (typically) smooth scalar function that has a regular zero at a bifurcation point. A bifurcation point between consecutive continuation points $x(s_k)$ and $x(s_{k+1})$ is *detected* when

$$\psi(x(s_k))\psi(x(s_{k+1})) < 0. \tag{5.1}$$

Once a bifurcation point has been detected, it can be *located* by solving a system of the form

$$\begin{cases} f(x) = 0, \\ g(x) = 0, \end{cases} \tag{5.2}$$

where g may be ψ or may be some other function which has a regular zero at the bifurcation point.

To detect Hopf points, MATCONT uses the test function

$$\psi(x(s)) := \det\left[2A(s) \odot I_n\right] = \prod_{i<j} \left(\lambda_i(s) + \lambda_j(s)\right), \tag{5.3}$$

where $\odot$ is the bialternate product [32]. Using the projection computed from the CIS algorithm, we introduce the analogous test function

$$\hat{\psi}(x(s)) := \det\left[2T_{11}(s) \odot I_m\right] = \prod_{i<j\le m} \left(\lambda_i(s) + \lambda_j(s)\right). \tag{5.4}$$

Clearly, $\psi(x(s))$ and $\hat{\psi}(x(s))$ are zero if $A(s)$ has a pure imaginary pair of eigenvalues $(\pm i\kappa)$, and so ψ and $\hat{\psi}$ can be used to test for Hopf bifurcations. However, these functions may also be zero because of a pair of real eigenvalues which sum to zero. Therefore, we also introduce a parity function which counts the number of unstable complex conjugate pairs:

$$\chi(x(s)) = (-1)^{\#\left\{\lambda_i(s):\mathrm{Re}\,\lambda_i(s)\ge 0 \text{ and } \mathrm{Im}\,\lambda_i(s)>0\right\}}. \tag{5.5}$$

We detect a Hopf bifurcation when

$$\hat{\psi}(x(s_k))\hat{\psi}(x(s_{k+1})) < 0 \text{ and } \chi(x(s_k))\chi(x(s_{k+1})) < 0. \tag{5.6}$$

A well-known method to locate a Hopf point (see, e.g., [36, 32, 5]) is to solve the system

$$\begin{cases} f(x) = 0, \\ f_u(x)r - i\omega r = 0, \\ r^* r_0 - 1 = 0, \end{cases} \tag{5.7}$$

where $x \in \mathbb{R}^{n+1}$, $r \in \mathbb{C}^n$, and $\omega \in \mathbb{R}$. The reference vector $r_0 \in \mathbb{C}^n$ is given. Usually, the system (5.7) is converted to a system of $3n+2$ real unknowns. Based on the CIS algorithm, we replace (5.7) with the system

$$\begin{cases} f(x) = 0, \\ T_{11}(x)r - i\omega r = 0, \\ r^* r_0 - 1 = 0, \end{cases} \tag{5.8}$$

where r and r_0 are now vectors in $\mathbb{C}^m$. In contrast to (5.7), the system (5.8) involves $n + 2m + 2$ real unknowns.

5.2. The one-dimensional Brusselator. The $1D$ *Brusselator* [37] is a well-known model system for autocatalytic chemical reactions with diffusion. The problem is defined on $\Omega = (0, 1)$ by coupled differential equations for unknowns u and v

$$\frac{d_1}{l^2}u'' - (b+1)u + u^2 v + a = 0,$$
$$\frac{d_2}{l^2}v'' + bu - u^2 v = 0$$

with boundary conditions

$$u(0) = u(1) = a \text{ and } v(0) = v(1) = \frac{b}{a}. \tag{5.9}$$

This problem exhibits a rich bifurcation scenario and has been used in the literature as a standard model for bifurcation analysis [39, 30, 15, 4, 13, 38]. Utilizing a second order finite difference discretization

$$f'' \approx \frac{1}{h^2}(f_{i-1} - 2f_i + f_{i+1}) \tag{5.10}$$

with $h = (N+1)^{-1}$, the resulting discrete problem can be written in the form (1.2). This discretization of the Brusselator is used in a MATCONT example [18].

In order to verify the accuracy of locating a Hopf point, we continue a constant solution branch: $u(x) = a$, $v(x) = \frac{b}{a}$, with respect to b. In this case the values of b where Hopf bifurcation occurs are known analytically as a function of N; see, e.g., [13, equation (24)]. Using MATLAB 7.0 on a 1.67 GHz G4, we located this bifurcation to at least eight correct digits for problems with $N = 1024$ to $N = 8192$ grid points; since there are two unknowns per grid point, the total size is $n = 2N$. Because these problems have only one-dimensional connectivity, the Jacobian may be reordered into a very narrowly banded form, and so the size to solve a linear system involving the Jacobian scales linearly with N. For each problem size, about 65% of the time was spent on spectrally transformed Arnoldi iterations using ARPACK; 12% of the time was spent on solving bordered systems for the corrector during continuation and for the Newton steps; and 7% of the time was spent on forming the Jacobian matrix. The cost of one Newton step for locating the bifurcation was approximately the same as the cost of one Newton step during the continuation, and to locate each bifurcation took three Newton steps. At $N = 8192$, the total time for fifteen steps of continuation and for locating one Hopf bifurcation was 158 seconds.

6. Conclusions and future work. In this paper, we have discussed the CIS algorithm for computing a smooth orthonormal basis for an invariant subspace of a parameter-dependent matrix, and we have extended it to make it more suitable for numerical bifurcation analysis. In particular, we have made the following contributions:

1. We have extended the original CIS algorithm for dense problems with logic for adapting the continued subspace in order to ensure that it always includes information relevant to bifurcation analysis. Such adaptation is necessary when a bifurcation occurs or when there is an *overlap*—that is, when the real parts of eigenvalues change order.
2. We have extended our algorithm to work efficiently on large sparse matrices by exploiting Galerkin projection methods. The original CIS algorithm used direct methods for dense matrices, and so cost $O(n^3)$ works at each step.
3. We have incorporated the projection-based CIS algorithm into the MATCONT bifurcation analysis package, and we have applied the combined code to the Brusselator model problem.

Future work includes the following topics. We are still actively investigating how the information can most effectively be used for finding bifurcations from non-static equilibria and how to best use the CIS algorithm in detecting and computing codimension-2 bifurcations along branches of Hopf and limit points. We are also involved in using the CIS algorithm in order to study the dependence of resonant frequencies of mechanical devices as design parameters are varied.

REFERENCES

[1] P.-A. Absil, R. Sepulchre, P. Van Dooren, and R. Mahony, *Cubically convergent iterations for invariant subspace computation*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 70–96.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, 3rd ed., SIAM, Philadelphia, 1999.

[3] P. Ashwin, K. Böhmer, and Z. Mei, *A numerical Liapunov-Schmidt method with applications to Hopf bifurcation on a square*, Math. Comp., 64 (1995), pp. 649–670, S19–S22.

[4] P. Ashwin and Z. Mei, *A Hopf bifurcation with Robin boundary conditions*, J. Dynam. Differential Equations, 6 (1994), pp. 487–503.

[5] W.-J. Beyn, A. Champneys, E. J. Doedel, Y. A. Kuznetsov, B. Sandstede, and W. Govaerts, *Numerical continuation and computation of normal forms*, in Handbook of Dynamical Systems III: Towards Applications, B. Fiedler, ed., Elsevier, Amsterdam, 2001, pp. 140–219.

[6] W.-J. Beyn, W. Kless, and V. Thümmler, *Continuation of low-dimensional invariant subspaces in dynamical systems of large dimension*, in Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems, B. Fiedler, ed., Springer, Berlin, 2001, pp. 47–72.

[7] D. Bindel, J. Demmel, and M. Friedman, *Continuation of invariant subspaces for large bifurcation problems*, in Proceedings of the SIAM Conference on Linear Algebra, Williamsburg, VA, 2003.

[8] D. Bindel, W. Demmel, M. Friedman, W. Govaerts, and Y. A. Kuznetsov, *Bifurcation analysis of large equilibrium systems in MATLAB*, in Proceedings of the ICCS conference 2005, Atlanta, GA, 2005, pp. 50–57.

[9] J. Bosec, *Continuation of Invariant Subspaces in Bifurcation Problems*, Ph.D. thesis, University of Marburg, Marburg, Germany, 2002.

[10] J. H. Brandts, *The Riccati algorithm for eigenvalues and invariant subspaces of matrices with inexpensive action*, Linear Algebra Appl. 358 (2003), pp. 335–365.

[11] E. A. Burroughs, R. B. Lehoucq, L. A. Romero, and A. J. Salinger, *Linear Stability of Flow in a Differentially Heated Cavity via Large-Scale Eigenvalue Calculations*, Technical report SAND2002-3036J, Sandia National Laboratories, Livermore, CA, 2002.

[12] C. S. Chien and M. H. Chen, *Multiple bifurcations in a reaction-diffusion problem*, Comput. Math. Appl., 35 (1998), pp. 15–39.

[13] C. S. Chien, Z. Mei, and C. L. Shen, *Numerical continuation at double bifurcation points of a reaction-diffusion problem*, Internat J. Bifur. Chaos Appl. Sci. Engrg., 8 (1997), pp. 117–139.

[14] K. A. Cliffe, A. Spence, and S. J. Tavener, *The numerical analysis of bifurcations problems vith application to fluid mechanics*, Acta Numer., 9 (2000), pp. 39–131.

[15] G. Dangelmayr, *Degenerate bifurcations near a double eigenvalue in the Brusselator*, J. Austral. Math. Soc. Ser. B, 28 (1987), pp. 486–535.

[16] J. W. Demmel, *Three methods for refining estimates of invariant subspaces*, Computing, 38 (1987), pp. 43–57.

[17] J. W. Demmel, L. Dieci, and M. J. Freidman, *Computing connecting orbits via an improved algorithm for continuing invariant subspaces*, SIAM J. Sci. Comput., 22 (2001), pp. 81–94.

[18] A. Dhooge, W. Govaerts, Y. Kuznetsov, W. Mestrom, and A. M. Riet, *MATLAB continuation software package CL_MATCONT*, Jan. 2003, available online at http://www.math.uu.nl/people/kuznet/cm/.

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

[20] L. Dieci and M. J. Friedman, *Continuation of invariant subspaces*, Numer. Linear Algebra Appl., 8 (2001), pp. 317–327.

[21] L. Dieci and A. Papini, *Continuation of eigendecompositions*, Future Generation Computer Systems, 19 (2003), pp. 1125–1137.

[22] L. Dieci and A. J. Rebaza, *Point-to-periodic and periodic-to-periodic connections*, BIT, 44 (2004), pp. 41–60.

[23] E. J. Doedel and H. Sharifi, *Collocation methods for continuation problems in nonlinear elliptic PDEs*, Notes Numer. Fluid Mech. 74 (2000), pp. 105–118.

[24] A. Edelman, T. Arias, and S. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 303–353.

[25] M. Friedman, W. Govaerts, Y. Kuznetsov, and B. Sautois, *Continuation of homoclinic orbits in MATLAB*, in Proceedings of the ICCS Conference 2005, Atlanta, GA, 2005, pp. 263–270.

[26] M. J. Friedman, *Improved detection of bifurcations in large nonlinear systems via the continuation of invariant subspaces algorithm*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 11 (2001), pp. 2277–2285.
[27] M. J. Friedman and M. E. Jackson, *An Improved RLV Stability Analysis via a Continuation Approach*, Technical, Report, NASA Marshall Space Flight Center, Huntsville, AL, 2002.
[28] K. Georg, *Matrix-free numerical continuation and bifurcation*, Numer. Funct. Anal. Optim., 22 (2001), pp. 303–320.
[29] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1989.
[30] M. Golubitsky and D. G. Schaeffer, *Singularities and Groups in Bifurcation Theory, Vol.* 1, Springer-Verlag, New York, 1985.
[31] W. Govaerts, *Computation of singularities in large nonlinear systems*, SIAM J. Numer. Anal., 34 (1997), pp. 867–880.
[32] W. Govaerts, *Numerical Methods for Bifurcations of Dynamical Equilibria*, SIAM, Philadelphia, 2000.
[33] W. Govaerts, J. Guckenheimer, and A. Khibnik, *Defining functions for multiple Hopf bifurcations*, SIAM J. Numer. Anal, 34 (1997), pp. 1269–1288.
[34] N. J. Higham, *Matrix nearness problems and applications*, in Applications of Matrix Theory, M. J. C. Gover and S. Barnett, eds., Oxford University Press, New York, 1989, pp. 1–27.
[35] T. Kato, *Perturbation Theory for Linear Operators*, Springer-Verlag, Corrected printing of the second edition, Springer-Verlag, Berlin, 1995.
[36] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory,* 2*nd ed.*, Springer-Verlag, New York, 1998.
[37] R. Lefever and I. Prigogine, *Symmetry-breaking instabilities in dissipative systems II*, J. Chem. Phys., 48 (1968), pp. 1695–1700.
[38] Z. Mei, *Numerical Bifurcation Analysis for Reaction-Diffusion Equations*, Ph.D. thesis, University of Marburg, Marburg, Germany, 1997.
[39] D. G. Schaeffer and M. A. Golubitsky, *Bifurcation analysis near a double eigenvalue of a model chemical reaction*, Arch. Rational Mech. Anal., 75 (1981), pp. 315–347.
[40] G. M. Shroff and H. B. Keller, *Stabilization of unstable procedures: The recursive projection method*, SIAM J. Numer. Anal, 30 (1993), pp. 1099–1120.
[41] G. W. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Rev., 15 (1973), pp. 727–764.
[42] G. W. Stewart and J.-G. Sun, *Matrix Perturbation Theory*, Academic Press, San Diego, CA, 1990.
[43] L. N. Trefethen, *Pseudospectra of matrices*, in Numerical Analysis 1991, D. F. Griffiths and G. A. Watson, eds., Longman Sci. Tech., Harlow, Essex, UK, 1991, pp. 234–262.
[44] J. M. Varah, *On the separation of two matrices*, SIAM J. Numer. Anal., 16 (1979), pp. 212–222.