Continuation of Invariant Subspaces for Large Bifurcation Problems

David Bindel [*]***, James Demmel*** [†]***, and Mark Friedman*** [‡]

Abstract

Recently, two of the authors and their collaborators have developed an algorithm for computing a smoothly varying basis for an invariant subspace of a parameter dependent matrix $A : \mathbb{R} \to \mathbb{R}^{n\times n}$. In this work, we extend the algorithm (i) to the case where n is large and direct methods are prohibitively expensive and (ii) to the case where eigenvalues collide, change order in the spectrum, or cross the imaginary axis.

1 Introduction

In the numerical analysis of dynamical systems parameter-dependent Jacobian matrices provide important information. For example, to analyze stability at steady state branches $(u(s), \alpha(s))$ of

$$\frac{d}{dt}u = f(u, \alpha),\ u \in \mathbb{R}^n,\ \alpha \in \mathbb{R},\ f(u, \alpha) \in \mathbb{R}^n \tag{1}$$

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, a low-dimensional invariant subspace $\mathcal{R}(s)$ of $A(s)$

[*]dbindel@eecs.berkeley.edu. Computer Science Division, University of California, Berkeley. This author was supported in part by an NSF Graduate Student Fellowship.

[†]demmel@eecs.berkeley.edu. Computer Science Division and Department of Mathematics, University of California, Berkeley

[‡]friedmam@math.uah.edu. Mathematical Sciences Department, University of Alabama, Huntsville. This author was supported in part under the 2002 NASA Faculty Fellowship Program at Marshall Space Flight Center and under NSF DMS-0209536.

corresponding to a small part of the spectrum near the imaginary axis provides information about stability and bifurcations.

Recently, two of the authors and their collaborator have developed the *CIS algorithm* for the continuation of invariant subspaces of a parameter dependent matrix $A(s) \in \mathbb{R}^{n\times n}$, $s \in [0,1]$. The CIS algorithm was presented and analyzed in [19] and further studied in [21], [25], with additional practical developments in [26].

In this paper we extend the CIS algorithm in two directions: (i) to the large and sparse case using Galerkin projection methods and (ii) to situations when the set $\Lambda_1(s)$ of eigenvalues of A restricted to $\mathcal{R}(s)$ collides with another eigenvalue, eigenvalues change order in the spectrum, and bifurcations occur.

Our principal goals are: (i) extending numerical bifurcation techniques developed for small and moderate size systems to large sparse systems and (ii) improving the reliability of detection of bifurcations. The latter is especially relevant for non-normal matrices, where a small perturbation may result in a large change to the eigenvalues [43, 44]. Our starting point was [33], in which a low-dimensional invariant subspace of the Jacobian matrix, corresponding to the leading eigenvalues, was computed at each continuation point and used for detecting 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 consists of a predictor based on first derivative information, and a corrector based on an iterative refinement technique for improving the accuracy of a computed invariant subspace (see [42], [18] and references therein). The algorithm samples a smoothly varying orthonormal basis $Q_1(s)$ for $\mathcal{R}(s)$ at points $0 = s_0 < s_1 < \ldots < s_{N-1} < s_N = 1$. The computed $Q_1(s)$ approximates a basis for $\mathcal{R}(s)$ with minimum arclength. The step size is chosen adaptively so that $h_i = s_i - s_{i-1}$ is decreased when $\mathcal{R}(s)$ changes fast and increased when $\mathcal{R}(s)$ changes slowly. When $\Lambda_1(s)$ comes too near to a complimentary set $\Lambda_2(s)$ of eigenvalues of A restricted to $\mathcal{R}(s)^{\perp}$, the size of the continued subspace is adapted, and continuation proceeds with a larger or smaller subspace.

We now briefly review some related work. In [23] the approach in [21] was extended to compute smooth block Schur factorizations in the case of more than two block of eigenvalues; and a new method was proposed to smoothly (block) bidiagonalize A. In [22] the algorithmic approach in [19] for connecting orbits between equilibria was extended to the case of the connecting orbits between periodic orbits.

A homotopy method for eigenpairs of large sparse matrices, which accounts for bifurcations and related phenomena, was presented in [37]. A continuation method for low-dimensional invariant subspaces of a parametrized family of large sparse matrices was proposed in a recent work [6], where a bordered version of the Bartels-Stewart algorithm was used at each continuation step. A continuation method for low-dimensional invariant subspaces which combines the approach in [6] with that in [21] and [25] was proposed in [8]. New algorithms for numerical approximation of eigenvalues and invariant subspaces of matrices with cheap action based on Galerkin projection of the algebraic Riccati equation were recently developed in [10].

Numerical continuation for large sparse nonlinear algebraic systems arising from discretization of ODEs and PDEs is an active research area. The continuation

algorithms are typically based on variants of Krylov subspace methods and recursive projection (RPM), see e.g. [2], [40], [31], [13], [32, Ch. 10], [24], [27], [12], [15].

2 Background

We will use script letters ($\mathcal{X}$) for spaces and corresponding Roman capitals (X) for bases. We will use the Frobenius matrix norm exclusively: $\|A\| = \sqrt{\mathrm{tr}(A^T A)}$. We write the eigenvalues of A as $\lambda_1, \ldots, \lambda_n$, where each eigenvalue with multiplicity k is listed k times. When A depends continuously on a parameter s, we will write the eigenvalues as possibly non-unique continuous functions $\lambda_1(s), \ldots, \lambda_n(s)$. We will write the spectrum of A as $\Lambda = \Lambda_1 \cup \Lambda_2$ where $\Lambda_1 = \{\lambda_i\}_{i=1}^m$ and $\Lambda_2 = \{\lambda_i\}_{i=m+1}^n$. We assume that complex conjugate pairs are not split between Λ_1 and Λ_2. When $\Lambda_1 \cap \Lambda_2 = \emptyset$, we will denote Λ_1's unique maximal right invariant subspace by $\mathcal{R}$.

Suppose that A is a C^k parameter-dependent matrix, and that at s_0 we have the block Schur decomposition

$$A(s_0) = \begin{bmatrix} Q_1(s_0) & Q_2(s_0) \end{bmatrix} \begin{bmatrix} T_{11}(s_0) & T_{12}(s_0) \\ 0 & T_{22}(s_0) \end{bmatrix} \begin{bmatrix} Q_1(s_0) & Q_2(s_0) \end{bmatrix}^T. \tag{2}$$

If $\Lambda_1(s_0) =$ spectrum of $T_{11}(s_0)$ is disjoint from $\Lambda_2(s_0)$, then we can constructively prove the existence of a continuous block Schur decomposition near s_0.

Theorem 1. *([19], [21]) Let A be a C^k matrix with a block Schur decomposition at s_0 as in (2), and let the spectra of $T_{11}(s_0)$ and $T_{22}(s_0)$ be disjoint. Let*

$$\widehat{T}(s) = \begin{bmatrix} \widehat{T}_{11}(s) & \widehat{T}_{12}(s) \\ E_{21}(s) & \widehat{T}_{22}(s) \end{bmatrix} := Q(s_0)^T A(s) Q(s_0). \tag{3}$$

Then for s sufficiently near s_0, there is a unique continuous minimum norm solution $Y(s) \in \mathbb{R}^{(n-m)\times m}$ to the algebraic Riccati equation

$$F(Y) := \widehat{T}_{22} Y - Y \widehat{T}_{11} + E_{21} - Y \widehat{T}_{12} Y = 0. \tag{4}$$

and there is a continuous block Schur decomposition

$$A(s) = Q(s) T(s) Q(s)^T \tag{5}$$

where

$$\bar{Q}(s) = Q(s_0) \begin{bmatrix} I & -Y(s)^T \\ Y(s) & I \end{bmatrix} \tag{6}$$

$$Q(s) = \bar{Q}(s) \begin{bmatrix} I + Y(s)^T Y(s) & 0 \\ 0 & I + Y(s) Y(s)^T \end{bmatrix}^{-1/2} \tag{7}$$

3 The CIS algorithm: direct case

We now describe the CIS algorithm in the case when we can use direct solvers. Much of this work is described in [19], [21], [25], and [26]. Here we emphasize aspects of the computation that we perform differently, or which are particularly relevant to the sparse case.

At the highest level our algorithm consists of the following steps: computation of an initial invariant subspace; a predictor-corrector iteration to continue the subspace from s_i to s_{i+1}; a normalization step; and logic to adapt the step size or reinitialize the space as needed. We can continue either $Q_1(s)$ and $T_{11}(s)$ alone, or we can continue the full $Q(s)$ and $T(s)$ matrices. Currently, the dense code computes full Schur factors. When we continue only the first part of the decomposition, as we do in the sparse case, we also need to compute a few eigenvalues from $\Lambda_2(s)$. We use $\Lambda_2(s)$ to decide whether to reinitialize the algorithm with a different partitioning of the spectrum.

3.1 Initialization

To initialize the algorithm at s_0, we compute a full Schur decomposition of $A(s_0)$ and use standard routines from LAPACK [1] to sort the Schur decomposition so that 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 are willing to include all the unstable eigenvalues as well as a few stable eigenvalues nearest the imaginary axis in our m-dimensional subspace.

We require that $\Lambda_1(s_0)$ contains all unstable eigenvalues and at least some specified number of stable eigenvalues, but we may wish to include additional eigenvalues to simplify subsequent continuation. For example, we will include an extra eigenvalue to avoid splitting a complex conjugate. More generally, we would like to choose $\Lambda_1(s_0)$ so that the gap between the real parts of $\Lambda_1(s_0)$ and $\Lambda_2(s_0)$ exceeds some threshold. Thus, we hope to avoid missing eigenvalues near the imaginary axis.

In the dense case, the same LAPACK routine used to sort the Schur form will also estimate the sensitivity of the selected subspace, and so we may also choose to use a larger subspace if it is very sensitive. Though the cost per step increases as we increase the size of our subspace, continuing a less sensitive subspace will allow longer steps.

3.2 Choosing a subspace

To compute $\mathcal{R}(s_1)$ from $\mathcal{R}(s_0)$, we use an Euler predictor followed by Newton corrections. We use the convergence rate of the Newton iteration to govern our step size. When Newton converges too slowly, we will reduce our step size, or possibly reinitialize the continuation process with a larger or smaller subspace. When Newton converges too quickly, we will increase our step size.

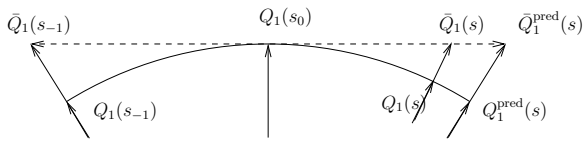

Figure 1. *Choosing a consistent normalization for secant prediction*

Subspace predictors

We can construct predictors in several ways. The simplest is to differentiate (4) and construct an Euler predictor. Since the Euler predictor does not generalize nicely to the sparse case, we derive a secant predictor. We can derive higher-order polynomial predictors in the same way.

In each predictor-corrector step, we normalize the basis for a space $\mathcal{X}$ by requiring that $Q(s_0)^T X = I$. If $\mathcal{R}(s_{-1})$ is the invariant subspace from a previous step, we must choose a basis for $\mathcal{R}(s_{-1})$ which is consistent with the current normalization (see Figure 3.2):

$$\bar{Q}_1(s_{-1}) = Q_1(s_{-1}) \left(Q_1(s_0)^T Q_1(s_{-1}) \right)^{-1} \tag{8}$$

In the context of the Riccati equation (4), we write this basis as

$$\bar{Q}_1(s_{-1}) = Q(s_0) \begin{bmatrix} I \\ Y(s_{-1}) \end{bmatrix} \text{ where } Y(s_{-1}) = Q_2(s_0)^T \bar{Q}_1(s_{-1}) \tag{9}$$

Now the secant predictor for the Riccati equation unknown Y is

$$Y_0(s_1) = -\frac{s_1 - s_0}{s_0 - s_{-1}} Y(s_{-1}) \tag{10}$$

Direct Newton corrector iterations

There are two ways to construct a Newton iteration for $\bar{Q}(s)$. One way is to write a residual equations for the eigensystem and for the normalization:

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

A Newton step for this system of equations can be solved using a bordered Bartels-Stewart algorithm [6]. Alternately, we can eliminate $\bar{T}_{11}(s_1)$ and perform Newton iteration on the Riccati equation (4). We solve a Newton step for the Riccati equation using an ordinary Bartels-Stewart algorithm [28, p. 367].

Newton iteration on the unreduced system of equations is equivalent to iteration on the reduced system, 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, the reduction to a Riccati equation, while it modestly reduces the total number of unknowns, destroys sparsity. For small problems we use dense methods, and sparsity loss matters

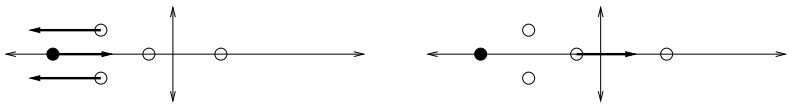

Figure 2. *Examples of overlap and bifurcation.*

little; for large problems we sidestep the issue by using projection methods. For intermediate size problems it may help to use sparse direct solvers to take Newton steps on the unreduced system of equations.

Normalizing the solution

After we compute a basis $\bar{Q}_1(s_1)$ for $\mathcal{R}(s_1)$, we normalize it to find another basis $Q_1(s_1)$ for $\mathcal{R}(s_1)$ which is as near as possible to $Q_1(s_0)$. This normalization approximates the minimal arclength condition $Q_1(s)^T Q_1'(s) = 0$ (see [11]). Let $\bar{Q}_1(s_1) = U\Sigma V^T$ be the SVD with Σ and V in square; then $Q_1(s_1) = UV^T$.

3.3 Subspace analysis and adaptation

Bifurcations and overlaps

When initialized at s_0, $\Lambda_1(s_0)$ contains all unstable eigenvalues of $A(s_0)$ and a few stable eigenvalues nearest the imaginary axis. $\Lambda_2(s_0)$ lies strictly to the left of $\Lambda_1(s_0)$ in the complex plane. During continuation, eigenvalues from $\Lambda_1(s)$ may cross the imaginary axis (a bifurcation; see Figure 2, right), or $\Lambda_2(s)$ may cease to lie strictly to the left of $\Lambda_1(s)$ (an overlap; see Figure 2, left). When bifurcation or overlap occurs, we reinitialize the continuation procedure.

A *generic* overlap or bifurcation persists when $A(s)$ 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. In three cases, a single real eigenvalue or complex conjugate pair from $\Lambda_2(s)$ enters the convex hull 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. In this fourth case, the orthonormal basis $Q_1(s)$ corresponding to $\Lambda_1(s)$ will cease to be continuously defined, and we expect that the Newton iteration will not converge. Complex conjugate eigenvalues 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.

Step size and subspace adaptation

We use the CIS algorithm with standard bifurcation analysis algorithms designed for small problems by restricting computations to a "small" matrix $T_{11}(s)$. Therefore,

we try to ensure that only eigenvalues from $\Lambda_1(s)$ can cross the imaginary axis. To prevent $\Lambda_2(s)$ from crossing the imaginary axis, we adapt the step size and the size of $\Lambda_1(s)$ so that overlaps and bifurcations do not occur in the same step.

When an overlap occurs because of two real eigenvalues colliding 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:

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

If $\mathcal{O}$ is non-empty, then an overlap has occurred. To identify multiple overlaps, we compute the cardinality of $\mathcal{O}$. To avoid double-counting overlaps involving complex conjugate pairs, we will only count the pairs such that $\mathrm{Im}(\lambda_i(s)) \leq 0$ and $\mathrm{Im}(\lambda_j(s)) \leq 0$.

Only a single overlap is allowed going from s_i to s_{i+1}. Hence if we detect multiple overlaps, we keep retrying with a smaller step size until only one overlap is left; we then accept this step. If we reach the minimum step size and still have multiple overlaps, we will reinitialize the continuation process at s_i so that $\mathcal{O}$ from the failed step belongs entirely to $\Lambda_1(s_i)$ or $\Lambda_2(s_i)$.

We detect bifurcations by counting unstable eigenvalues. If the number differs between s_i and s_{i+1}, then a bifurcation occurred during the step. If the number changed by more than one real eigenvalue or one conjugate eigenpair, we assume that a multiple bifurcation has occurred, and we try to resolve it by decreasing the step size. If we cannot resolve the behavior with the minimum step size, then we fail 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}, 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: iterative case

We now turn to the case when the dimension n of $A(s)$ is large and $m \ll n$. In this case, direct methods are expensive, so we will use projection methods.

4.1 Choosing a projection space

In the dense case, we consider 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 sparse case, we consider three spectral sets: $\Lambda_1(s)$, a set of m elements which contains the unstable eigenvalues and a few of the rightmost stable eigenvalues; $\Lambda_2(s)$, a set of $p - m$ elements which contains a few of the rightmost eigenvalues not in $\Lambda_1(s)$; and $\Lambda_3(s)$, a set of $n - p$ elements which contains the remainder of the spectrum. Our basic strategy in the projected CIS algorithm is to build a projection space $\mathcal{V}$ of dimension $p > m$ such that the

restriction of $A(s)$ to $\mathcal{V}$ provides good approximations to $\Lambda_1(s)$ and $\Lambda_2(s)$. Our current code builds a projection space $\mathcal{V}$ by using the implicitly restarted Arnoldi code ARPACK [41] together with a spectral transformation.

4.2 Initialization

During initialization, we may not know how many large $\mathcal{V}$ must be. Therefore, the sparse version of the initialization routine calls the dense initialization code in a loop with ever larger projection spaces. If not enough stable eigenvalues converge or there are no sufficiently large gaps between stable eigenvalues in the converged part of the spectrum, more eigenvalues are requested. If a suitable subspace cannot be found when a specified maximum number of eigenvalues are requested, the code exits with a diagnostic message.

4.3 Projected prediction and correction

Suppose $V \in \mathbb{R}^{p\times n}$ is an orthonormal basis for a projection space $\mathcal{V}$. Then we approximate the equation

$$A(s)\bar{Q}_1(s) - \bar{Q}_1(s)\bar{T}_{11}(s) = 0 \tag{12}$$

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

$$V^T A(s) V\hat{Q}_1(s) - \hat{Q}_1(s)\bar{T}_{11}^h(s) = 0 \tag{13}$$

Once $\bar{Q}_1^h(s)$ has been computed, we normalize as in the dense case to find $Q_1^h(s)$ spanning the same space which is as near as possible to $Q_1(s_0)$ in 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 span($Q_1^h(s)$ in $\mathcal{V}$.

We typically will use a projection space $\mathcal{V}$ which is itself a computed maximal invariant subspace. 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 (13) *exactly* 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 seek to compute the upper left part of a three-by-three block Schur form

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

where $Q(s_1) = \begin{bmatrix} Q_1^h(s_1) & Q_2^h(s_1) & V^\perp \end{bmatrix}$. The spectrum of the $T_{11}^h(s_1)$ block will be the continued set of eigenvalues $\Lambda_1(s_1)$. The $T_{22}^h(s_1)$ block will have a few of the rightmost remaining eigenvalues, which we use to diagnose overlap. The eigenvalues of the uncomputed block $T_{33}^h(s_1)$ are part of the spectrum which lies further away from the imaginary axis.

As in the dense case, we can eliminate $\bar{T}_{11}^h(s)$ from equation (13) to get a projected Riccati equation whose minimum-norm solution corresponds to $\mathcal{R}(s)$ [7].

We solve the projected Riccati equation by Newton iteration. To get a Newton starting point at s_1, we compute a secant prediction $\bar{Q}_1^{\text{pred}}(s_1)$ as in the dense case, and then use the projection $\bar{Q}_1^{h,\text{pred}}(s_1) = VV^T\bar{Q}_1^{\text{pred}}(s_1)$.

5 Integrating the CIS algorithm into MATCONT

We are incorporating our CIS code into *CL_MATCONT* [20], a bifurcation software package in MATLAB. We consider one example: Hopf bifurcation.

Let $x(s) = (u(s), \alpha(s))$ be a smooth local parametrization of a solution branch to the stationary problem (1), and let $A(s) := f_u(x(s))$ have eigenvalues $\lambda_i(s)$. We say $x_0 = (u_0, \alpha_0) = (u(s_0), \alpha(s_0))$ is a *bifurcation point* if at least one $\lambda_i(s_0)$ has zero real part. It is a simple *Hopf* bifurcation point (H) if a complex conjugate eigenpair λ of $A(s)$ satisfies: (i) $\operatorname{Re}\lambda(s_0) = 0$ and (ii) $(\operatorname{Re}\lambda(s_0))' \neq 0$.

A *test function* $\phi(s) := \psi(x(s))$ is a smooth scalar function that has a regular zero at a bifurcation point. A bifurcation point is *detected* when $\psi(x(s))$ changes sign between two consecutive points. It is usually *located* by applying Newton's method to the system $f(x) = 0,\ \psi(x) = 0$.

By default, *MATCONT* uses the test function

$$\psi_H^M(x(s)) = \det\left[2f_u(x(s)) \odot I_n\right] \tag{14}$$

to detect Hopf bifurcations. To locate Hopf bifurcations, *MATCONT* finds zeros of $\psi_H^M(x(s))$ by a secant method. The bi-alternate product $2f_u(x(s)) \odot I_n$ has dimension $n(n-1)/2$-by-$n(n-1)/2$; its eigenvalues are the pairwise sums of the eigenvalues of $f_u(x(s))$.

To detect Hopf points, we introduce the following test functions:

$$\psi_H^1(x(s)) = \prod_{m \geq i > j} \left(\operatorname{Re}\lambda_i(s) + \operatorname{Re}\lambda_j(s)\right), \tag{15}$$

$$\psi_H^2(x(s)) = \begin{cases} 1, & \dim\left(\Lambda^{c+}(s_k)\right) = \dim\left(\Lambda^{c+}(s_{k+1})\right) \text{ and} \\ & \dim\left(\Lambda^{c-}(s_k)\right) = \dim\left(\Lambda^{c-}(s_{k+1})\right), \\ -1, & \text{otherwise,} \end{cases} \tag{16}$$

where $\Lambda^{c+}(s)$ and $\Lambda^{c-}(s)$ are the ordered subsets of $\Lambda_1(s)$ consisting of complex eigenvalues with $\operatorname{Re}(\lambda_i(s)) \geq 0$ and $\operatorname{Re}(\lambda_i(s)) < 0$, respectively. The condition for a Hopf bifurcation is

$$\mathrm{H} : \psi_H^1\left(x(s_k)\right)\psi_H^1\left(x(s_{k+1})\right) < 0 \text{ and } \psi_H^2\left(x(s_k)\right)\psi_H^2\left(x(s_{k+1})\right) < 0. \tag{17}$$

Because we restrict these test functions to the part of the spectrum computed by the CIS algorithm, it costs us relatively little to evaluate them.

Once a Hopf bifurcation has been detected, we use an extended system to locate the bifurcation. Hopf points are regular solutions of

$$\left\{ \begin{array}{c} f(x) = 0, \\ f_u(x)r - i\omega r = 0, \\ r^* r_0 - 1 = 0 \end{array} \right\},\ x \in \mathbb{R}^n,\ \omega \in \mathbb{C},\ r \in \mathbb{C}^n. \tag{18}$$

A standard method of locating Hopf points is to solve (18) by Newton's method (see e.g. [36], [32], [5]). Using the basis $Q_1(x(s))$ computed by the CIS algorithm, we replace f_u with $Q_1^T f_u Q_1$ and replace r with $Q_1\hat{r}, \hat{r} \in \mathbb{C}^m$. While (18) has $3n+2$ unknowns, the reduced system has only $n+2m+2$ unknowns.

Example 1. *The* 1D Brusselator*, a well known model system for autocatalytic chemical reactions with diffusion, is given by:*

$$\begin{array}{ll} \frac{d_1}{l^2}u'' - (b+1)u + u^2 v + a = 0, & \frac{d_2}{l^2}v'' + bu - u^2 v = 0, \quad in\ \Omega = (0,1), \\ u(0) = u(1) = a, & v(0) = v(1) = \frac{b}{a}. \end{array} \tag{19}$$

This problem exhibits many interesting bifurcations and has been used in the literature as a standard model for bifurcation analysis (see e.g. [39], [16], [17], [3], [14], and [38]). We discretize the problem with a standard second-order finite difference approximation for the second derivatives at N mesh points. We write the resulting system, which has dimension $n = 2N$, in the form (1).

To verify that our prototype code accurately locates Hopf points, we continue a constant solution branch of the Brusselator. For this problem, the parameter values b for which Hopf bifurcations occur are known analytically as a function of the discretization density N (see e.g. [14, Eq. (24)]). In the table below, $n = 2N$ is the dimension of the system (1); t_{total} and t_{CIS} are the average time time taken for a continuation step by *MATCONT* and the part of the time spent in our prototype CIS code. The average time to locate Hopf points with at least 8 correct digits for the parameter value is t_H. The computations are performed on an 850 MHz Dell Inspiron 8000 running Matlab 6.1. The continued subspace was initially dimension 8, but grew to dimension 11 over the course of the continuation independent of n.

n	256	512	1024	2048	4096
$t_{total}\frac{\text{sec.}}{step}$	1.13	1.56	2.20	4.35	13.8
$t_{CIS}\frac{\text{sec.}}{step}$	1.01	1.19	1.25	1.29	1.41
t_H sec.	9.1	9.5	10.7	18.8	39

For this problem, the time taken by the CIS algorithm remains largely steady, and the cost is dominated by the Newton iteration in *MATCONT*. We believe this is due to the way in which *MATCONT* solves the extended systems used in continuation; by modifying *MATCONT*'s solution algorithm to use a mixed block elimination procedure [30, 34], we believe we will be able to reduce the cost of a *MATCONT* step. Since we followed a constant branch, *MATCONT* only made one Newton correction per continuation step. Locating a Hopf bifurcation typically takes three Newton iterations, each of which is as costly as a *MATCONT* Newton correction.

Bibliography

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third ed., 1999.

[2] 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 and S19–S22.

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

[4] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, eds., *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[5] W.-J. Beyn, A. Champneys, E. 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, 2001, ch. 4.

[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, 2001, pp. 47 – 72.

[7] D. Bindel, J. Demmel, and M. Friedman, *Continuation of invariant subspaces for large bifurcation problems.* Technical report in progress, 2003.

[8] J. Bosec, *Continuation of Invariant Subspaces in Bifurcation Problems*, PhD thesis, University of Marburg, 2002.

[9] J. Brandts, *Matlab code for sorting real Schur forms*, Numerical Linear Algebra Applications, 9 (2002), pp. 249–261.

[10] ——, *The Riccati method for eigenvalues and invariant subspaces of matrices with inexpensive action*, Linear Algebra and its Applications, 358 (2003), pp. 335–365.

[11] A. Bunse-Gerstner, R. Byers, V. Mehrmann, and N. Nichols, *Numerical computation of an analytic singular value decomposition of a matrix valued function*, Numerische Mathematik, 60 (1991), pp. 1–39.

[12] E. Burroughs, R. Lehoucq, L. Romero, and A. Salinger, *Linear stability of flow in a differentially heated cavity via large-scale eigenvalue calculations*, Tech. Rep. SAND2002-3036J, Sandia National Laboratories, 2002.

[13] C. Chien and M. Chen, *Multiple bifurcations in a reaction-diffusion problem*, Computers Math. Applic., 35 (1998), pp. 15–39.

[14] C. Chien, Z. Mei, and C. Shen, *Numerical continuation at double bifurcation points of a reaction-diffusion problem*, Int. J. Bifur. and Chaos, 8 (1997), pp. 117–139.

[15] K. Cliffe, A. Spence, and S. Tavener, *The numerical analysis of bifurcations problems with application to fluid mechanics*, Acta Numerica, (2000), pp. 1–93.

[16] M. Colubitsky and D. Schaeffer, *Singularities and groups in bifurcation theory, Vol 1*, Springer-Verlag, 1985.

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

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

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

[20] A. Dhooge, W. Govaerts, Y. Kuznetsov, W. Mestrom, and A. Riet, *MATLAB continuation software package CL_MATCONT*, January 2003. `http://www.math.uu.nl/people/kuznet/cm/`.

[21] L. Dieci and M. Friedman, *Continuation of invariant subspaces*, Numerical Linear Algebra Applications, 8 (2001), pp. 317–327.

[22] L. .Dieci and A. Papini, *Point-to-periodic and periodic-to-periodic connections*. Preprint, 2003.

[23] L. Dieci and A. Rebaza, *Continuation of eigendecompositions*, To appear in FGCS: Future Generation Computer Systems – Computational Science, (2002).

[24] E. Doedel and H. Sharifi, *Collocation methods for continuation problems in nonlinear elliptic PDEs, issue on continuation*, 74 (2000), pp. 105–118.

[25] M. Friedman, *Improved detection of bifurcations in large nonlinear systems via the Continuation of Invariant Subspaces algorithm*, Int. J. Bif. and Chaos, 11 (2001), pp. 2277–2285.

[26] M. Friedman and M. Jackson, *An improved RLV stability analysis via a continuation approach*, tech. rep., NASA Marshall Space Flight Center, 2002.

[27] K. Georg, *Matrix-free numerical continuation and bifurcation*, Numerical Functional Analysis and Optimization, 22 (2001), pp. 303–320.

[28] G. Golub and C. V. Loan, *Matrix Computations*, The John Hopkins University Press, 1989.

[29] G. Golub, S. Nash, and C. F. V. Loan, *A Hessenberg-Schur method for the problem $AX - XB = C$*, IEEE Trans. Automat. Control., (1979), pp. 909–913.

[30] W. Govaerts, *Stable solvers and block elimination for bordered systems*, SIAM J. Matrix Anal. Appl., 12 (1991), pp. 469–483.

[31] ———, *Computation of singularities in large nonlinear systems*, SIAM J. Numer. Anal., 34 (1997), pp. 867–880.

[32] ———, *Numerical methods for bifurcations of dynamical equilibria*, Siam Publications, 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] W. Govaerts and J. D. Pryce, *Mixed block elimination for linear systems with wider borders*, IMA J. Numer. Anal., 13 (1993), pp. 161–180.

[35] D. Hu and L. Reichel, *Krylov subspace methods for the Sylvester equations*, Linear Algebra and its Applications, (1992), pp. 283–314.

[36] Y. A. Kuznetsov, *Elements of Applied Bifurcation Theory, Second edition*, Springer-Verlag, New York, 1998.

[37] S. Lui, H. Keller, and T. Kwok, *Homotopy method for the large, sparse, real nonsymmetric eigenvalue problem.*, SIAM J. Matrix Anal, 18 (1997), pp. 312–333.

[38] Z. Mei, *Numerical bifurcation analysis for reaction-diffusion equations*, PhD thesis, University of Marburg, 1997.

[39] D. Schaeffer and M. Golubitsky, *Bifurcation analysis near a double eigenvalue of a model chemical reaction*, Arch. Rational Mech. Anal., 75 (1981), pp. 315–347.

[40] G. Shroff and H. Keller, *Stabilization of unstable procedures: The recursive projection method*, SIAM J. Numer. Anal, 30 (1993), pp. 1099–1120.

[41] D. Sorensen, *Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations*, 1996.

[42] G. Stewart, *Error and perturbation bounds for subspaces associated with certain eigenvalue problems*, SIAM Review, 4 (1973), pp. 727–764.

[43] L. Trefethen, *Pseudospectra of matrices*, in Numerical Analysis 1991, D. Griffiths and G. Watson, eds., Longman Scientific and Technical, Harlow, Essex, UK, 1991, pp. 234–262.

[44] J. Varah, *On the separation of two matrices*, SIAM J. Num. Anal., (1979), pp. 212–222.