

On the location and continuation of Hopf bifurcations in large-scale problems.

M. Friedman * W. Qiu †

March 6, 2007

Abstract

CL_MATCONT is a MATLAB package for the study of dynamical systems and their bifurcations. It uses a *minimally augmented system* for continuation of the Hopf curve. The Continuation of Invariant Subspaces (CIS) algorithm produces a smooth orthonormal basis for an invariant subspace $\mathcal{R}(s)$ of a parameter-dependent matrix $A(s)$. We extend a minimally augmented system technique for location and continuation of Hopf bifurcations to large-scale problems using the *CIS algorithm*, which has been incorporated into CL_MATCONT. We compare this approach with using a *standard augmented system* and show that a minimally augmented system technique is more suitable for large-scale problems. We also suggest an improvement of a minimally augmented system technique for the case of the torus continuation.

1 Introduction.

CL_MATCONT is a MATLAB

Consider a parametrized dynamical system defined as a set of ODEs of the form

$$\frac{du}{dt} = f(u, \alpha), \quad (1)$$

where $u \in \mathbb{R}^n$ is a *state variable*, $\alpha \in \mathbb{R}$ is a *parameter*, and $f(u, \alpha) \in \mathbb{R}^n$ is a smooth function of u and α . For a general background on numerical continuation we refer to the existing literature; see e.g., [1], [15], [16], [21, Section 3.6], [3], [22].

*Mathematical Sciences Department, University of Alabama in Huntsville, Huntsville, AL 35899, friedman@math.uah.edu. The author was supported in part under NSF DMS-0209536 and NSF ATM-0417774.

†Mathematical Sciences Department, University of Alabama in Huntsville, Huntsville, AL 35899, frederick_qiu@yahoo.com. The author was supported in part under NSF DMS-0209536.

We are interested in locating generic codimension-1 *Hopf* bifurcation points on branches $x(s) \equiv (u(s), \alpha(s))$ of steady states

$$f(x) \equiv f(u, \alpha) = 0 \tag{2}$$

and in two parameter continuation of the Hopf curve. Specifically, we are interested in the case of large stationary problems (2), when direct methods are expensive. However, if we can multiply by the Jacobian matrix f_u quickly, we can use projection methods. A typical example comes from a spatial discretization of elliptic partial differential equations, in which case f_u 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.

CL_MATCONT [13] and its GUI version MATCONT [12] are MATLAB packages for the study of dynamical systems and their bifurcations for small and moderate size problems. They use *minimally augmented systems* (see [21], [13]) for continuation of the Hopf curve.

The CIS algorithm is an algorithm for computing a smooth orthonormal basis for an invariant subspace $\mathcal{R}(s)$ of a parameter-dependent matrix $A(s)$ [10, 14, 19, 5], [6]. This allows for computation and continuation of a much smaller restriction $C(s) := A(s)|_{\mathcal{R}(s)}$ of $A(s)$ onto $\mathcal{R}(s)$. And $C(s)$ provides all the relevant information about bifurcations. The CIS algorithm uses projection methods to deal with large problems. See also [4] for similar results.

We have incorporated the CIS algorithm (in this case $A(x(s)) = f_u(x(s))$) into CL_MATCONT to extend its functionality to large-scale bifurcation computations via subspace reduction, see [5], [7], [18] for some initial results. In the present paper we extend the minimally augmented systems technique to large-scale problems for location and continuation of Hopf bifurcations. We are preparing a detailed paper which will have a complete description of the extension of the functionality of CL_MATCONT to large-scale bifurcation problems.

For locating a Hopf bifurcation, we use both a *minimally augmented system* and a standard augmented system, and compare the performance of these two methods. We use the BED (Block Elimination Doolittle) algorithm to solve the resulting linear systems. In addition, we use a modification of BED to construct a well-conditioned minimally augmented system. Using the same approach, we also suggest an improvement of a minimally augmented system technique in the case of the torus continuation.

For continuation of a Hopf bifurcation, we use a minimally augmented system only. A standard augmented system is not suitable in this case, since the number of the equations there depends on the size of $C(x)$ which may change during continuation of a Hopf bifurcation, but not during location of a Hopf bifurcation.

2 Preliminaries.

2.1 Subspace reduction for large systems.

Let $A(s) \in \mathbb{R}^{n \times n}$, $s \in [0, 1]$, be a C^k parameter dependent matrix (in our case $A(s)$ is the Jacobian matrix $f_u(x(s))$). For some $m \ll n$, let

$$\Lambda_1(s) \equiv \{\lambda_i(s)\}_{i=1}^m, \operatorname{Re} \lambda_m \leq \dots \leq \operatorname{Re} \lambda_{m_u+1} < 0 \leq \operatorname{Re} \lambda_{m_u} \leq \dots \leq \operatorname{Re} \lambda_1, \quad (3)$$

be a small set consisting of rightmost eigenvalues of $A(s)$ and let $Q_1(s) \in \mathbb{R}^{n \times m}$ be an orthonormal basis for the invariant subspace $\mathcal{R}(s)$ corresponding to $\Lambda_1(s)$. Then an application of the CIS algorithm [10, 14, 19, 5], [7], [6] to $A(s)$ produces

$$C(s) \equiv T_{11}(s) = Q_1^T(s)A(s)Q_1(s) \in \mathbb{R}^{m \times m}, \quad (4)$$

which is the restriction of $A(s)$ onto $\mathcal{R}(s)$. Here T_{11} comes from the block Schur decomposition

$$A(s) = [Q_1(s) \quad Q_2(s)] \begin{bmatrix} T_{11}(s) & T_{12}(s) \\ 0 & T_{22}(s) \end{bmatrix} [Q_1(s) \quad Q_2(s)]^T. \quad (5)$$

Moreover, the CIS algorithm ensures that the only eigenvalues of $A(s)$ that can cross the imaginary axis come from $\Lambda_1(s)$, and these are exactly the eigenvalues of $C(s)$. We use this result to construct new methods for detecting and locating bifurcations. Note, that $\Lambda_1(s)$ is computed automatically whenever $C(s)$ is computed.

2.2 Solving bordered systems: BED algorithm.

Consider linear systems of the form (see [21, Section 3.6])

$$M \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \quad (6)$$

where M has the bordered form

$$M = \begin{bmatrix} A & B \\ C^T & D \end{bmatrix}, \quad (7)$$

with $A \in \mathbb{R}^{n \times n}$ is large and sparse, $B, C \in \mathbb{R}^{n \times k}$, $D \in \mathbb{R}^{k \times k}$, $x, f \in \mathbb{R}^n$, $y, g \in \mathbb{R}^k$, and $k \ll n$. In case when both M and A are well conditioned, (6) can be solved accurately using a block LU factorization of M by the BED (block elimination Doolittle) algorithm, based on the factorization

$$\begin{bmatrix} A & B \\ C^T & D \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ \bar{C}^T & I_k \end{bmatrix} \begin{bmatrix} A & B \\ 0 & \bar{D} \end{bmatrix}. \quad (8)$$

Algorithm 1 *BED (Block Elimination Doolittle) algorithm.*

Step 1 Solve

$$A^T \bar{C} = C,$$

Step 2 Compute

$$\bar{D} \equiv D - \bar{C}^T B,$$

Step 3 Solve

$$\bar{D} y = g - \bar{C}^T f, \quad (9)$$

Step 4 Solve.

$$Ax = f - By.$$

Remark 1 We use SVD, which has better stability properties than LU factorization, to solve small linear system (9).

Remark 2 We use the BED algorithm to solve linear systems arising in the situation when a minimally augmented system is used for location of a Hopf bifurcation and for continuation of a Hopf bifurcation. In both these cases, A is usually well-conditioned, but \bar{D} can be ill-conditioned if we are not careful with our choice of the minimally augmented system. We hence consider matrices M_i with different borders and use Steps 1 and 2 of Algorithm 1 to choose the index i so that \bar{D}_i has the smallest condition number. This computation is summarized in Algorithm 2.

Algorithm 2 Select a bordered matrix M_i so that the corresponding \bar{D}_i has the smallest condition number.

Input: $M_i := \begin{bmatrix} A & B \\ C_i^T & D_i \end{bmatrix}$ nonsingular where $A \in \mathbb{R}^{n \times n}$ and nonsingular, $B \in \mathbb{R}^{n \times k}$, $C_i \in \mathbb{R}^{n \times k}$, $D_i \in \mathbb{R}^{k \times k}$ and $1 \leq i \leq K$.

Step 1 Solve the linear system:

$$A^T \begin{bmatrix} \bar{C}_1 & \cdots & \bar{C}_K \end{bmatrix} = \begin{bmatrix} C_1 & \cdots & C_K \end{bmatrix} \quad (10)$$

Step 2 Compute

$$\begin{bmatrix} \bar{D}_1 & \cdots & \bar{D}_K \end{bmatrix} = \begin{bmatrix} D_1 & \cdots & D_K \end{bmatrix} - \begin{bmatrix} \bar{C}_1^T & \cdots & \bar{C}_K^T \end{bmatrix} B. \quad (11)$$

Step 3 Compute the condition numbers of $\bar{D}_1, \dots, \bar{D}_K$.

Output: M_i, i , where \bar{D}_i has the smallest condition number.

Remark 3 At Step 1 we solve a linear system with $k \cdot K$ right hand sides, where $k \cdot K \ll n$. Specifically, $k = 2$ and $K = 6$ in the case of locating Hopf, where M_i are the Jacobian matrices (20) below; $k = 3$ and $K = 6$ in the case of continuation of a Hopf bifurcation (the size of the border is increased by one). Hence in both cases these computations are cheap.

3 Locating a Hopf bifurcation.

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 system (2). We write the Jacobian matrix along this path as $A(x(s)) \equiv f_u(x(s))$. Let $x_0 = x(s_0)$ be a simple Hopf point on a branch $x(s)$ of the system (2). This means that $A^0 \equiv A(x_0)$ has an imaginary conjugate eigenpair $\lambda = \pm i\omega$, $\omega > 0$, $\kappa = \omega^2$, and that $d(\operatorname{Re} \lambda)/ds \neq 0$ at s_0 .

We will also use the notation $A \equiv A(x)$, $f_x \equiv f_x(x)$, and, for restriction $C(x)$ of A onto $\mathcal{R}(s)$ defined by (4), $C \equiv C(x)$, $C^0 \equiv C(x_0)$.

We use a *standard augmented system* only for locating a Hopf point, while we use a *minimally augmented system* for locating a Hopf point and for continuation of a Hopf point. We do not consider the case of continuation of Hopf points separately, since the equations in this case are very similar to those used for locating a Hopf point, the main difference being that $\alpha \in \mathbb{R}^2$, and an additional scalar equations is added to (2).

3.1 Locating a Hopf bifurcation using a standard augmented system.

A well known method to locate a Hopf point (see e.g. [21], [3], [22]) is by applying Newton's method to a *standard augmented system* of $3n + 2$ scalar equations for $3n + 2$ components $(x, \hat{p}, \hat{q}, \omega) \in \mathbb{R}^{n+1} \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}$, $x = (u, \alpha) \in \mathbb{R}^n \times \mathbb{R}$,

$$\begin{cases} f(x) = 0, \\ f_u(x)\hat{p} + \omega\hat{q} = 0, \\ f_u(x)\hat{q} - \omega\hat{p} = 0, \\ \hat{p}_0^T \hat{p} + \hat{q}_0^T \hat{q} - 1 = 0, \\ -\hat{q}_0^T \hat{p} + \hat{p}_0^T \hat{q} = 0, \end{cases} \quad (12)$$

where $\hat{p}_0, \hat{q}_0 \in \mathbb{R}^n$ are certain fixed vectors. The Jacobian matrix for the system (12) is:

$$\hat{J} = \begin{bmatrix} f_x & 0 & 0 & 0 \\ f_{ux}\hat{p} & f_u & \omega I_n & \hat{q} \\ f_{ux}\hat{q} & -\omega I_n & f_u & -\hat{p} \\ 0^T & \hat{p}_0^T & \hat{q}_0^T & 0 \\ 0^T & -\hat{q}_0^T & \hat{p}_0^T & 0 \end{bmatrix} \in \mathbb{R}^{(3n+2) \times (3n+2)}. \quad (13)$$

Note, that (12) is the real form of the complex system for (x, \hat{r}, ω)

$$\begin{cases} f(x) = 0, \\ f_u(x)\hat{r} - i\omega\hat{r} = 0, \\ \hat{r}^T \hat{r}_0 - 1 = 0, \end{cases} \quad (14)$$

which defines a necessary condition for Hopf bifurcation, and a Hopf point x corresponds to a regular solution of (12). Here $\hat{r} = \hat{p} + i\hat{q}$, $\hat{r}_0 = \hat{p}_0 + i\hat{q}_0 \in \mathbb{C}^n$.

Using the CIS algorithm, the equations (12) are replaced, [5] and [6], by only $n+2m+2$ equations for $n+2m+2$ components $(x, p, q, \omega) \in \mathbb{R}^{n+1} \times \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}$,

where m is the size of $C(x)$:

$$\begin{cases} f(x) = 0, \\ C(x)p + \omega q = 0, \\ C(x)q - \omega p = 0, \\ p_0^\top p + q_0^\top q - 1 = 0, \\ -q_0^\top p + p_0^\top q = 0, \end{cases} \quad (15)$$

where $p_0, q_0 \in \mathbb{R}^m$ are certain fixed vectors. The Jacobian matrix for the system (15) is:

$$J = \begin{bmatrix} f_x & 0 & 0 & 0 \\ C_x p & C & \omega I_m & q \\ C_x q & -\omega I_m & C & -p \\ 0^\top & p_0^\top & q_0^\top & 0 \\ 0^\top & -q_0^\top & p_0^\top & 0 \end{bmatrix} = \begin{bmatrix} A & f_\alpha & 0 & 0 & 0 \\ C_u p & C_\alpha p & C & \omega I_m & q \\ C_u q & C_\alpha q & -\omega I_m & C & -p \\ 0^\top & 0 & p_0^\top & q_0^\top & 0 \\ 0^\top & 0 & -q_0^\top & p_0^\top & 0 \end{bmatrix} \in \mathbb{R}^{(n+2m+2) \times (n+2m+2)}, \quad (16)$$

where $f_x \in \mathbb{R}^{n \times (n+1)}$, $C_x(x)p, C_x(x)q \in \mathbb{R}^{m \times (n+1)}$. To reduce the cost of computing J , we use an approximation

$$C_x p = (Q_1^\top(x) A(x) Q_1(x))_x p \approx Q_1^\top(x) A_x(x) Q_1 p,$$

which is accurate when the subspace $\mathcal{R}(s)$ does not vary much at $x : (Q_1(x))_x$ is 'small' and a first order finite difference approximation

$$A_x(x)v = (f_x)_u v = \frac{f_x(u + \delta \frac{v}{\|v\|}, \alpha) - f_x(u)}{\delta} \|v\| + O(\delta) \quad v \in \mathbb{R}^n.$$

Combining the above two equations, we get

$$C_x(x)p \approx Q_1^\top \frac{f_x(u + \delta \frac{v}{\|v\|}, \alpha) - f_x(u)}{\delta} \|v\|, \quad v = Q_1 p \in \mathbb{R}^n. \quad (17)$$

Similarly,

$$C_x(x)q \approx Q_1^\top \frac{f_x(u + \delta \frac{v}{\|v\|}, \alpha) - f_x(u, \alpha)}{\delta} \|v\|, \quad v = Q_1 q. \quad (18)$$

The system (15) with the Jacobian matrix J (16), (17), (18), is solved by the Newton method. And at each Newton step linear systems with the matrix J are solved by BED, since both J and $A \in \mathbb{R}^{n \times n}$ are well conditioned.

3.2 Locating Hopf using a minimally augmented system.

The minimally augmented system (see [21], [12], [13]) to locate x_0 consists of $n + 2$ scalar equations for $n + 2$ components $(x, \kappa) \equiv (u, \alpha, \kappa) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}$, $(i_1, j_1, i_2, j_2) \in \{1, 2\}$:

$$\begin{cases} f(x) = 0, \\ g_{i_1 j_1}(x, \kappa) = 0, \\ g_{i_2 j_2}(x, \kappa) = 0. \end{cases} \quad (19)$$

The Jacobian matrix of system (19) is:

$$\begin{bmatrix} f_u & f_\alpha & 0_{n \times 1} \\ (g_{i_1 j_1})_u & (g_{i_1 j_1})_\alpha & (g_{i_1 j_1})_\kappa \\ (g_{i_2 j_2})_u & (g_{i_2 j_2})_\alpha & (g_{i_2 j_2})_\kappa \end{bmatrix} \in \mathbb{R}^{(n+2) \times (n+2)}, \quad (20)$$

where $g_{ij} \equiv g_{ij}(x, \kappa)$.

Remark 4 Depending on possible choices of $g_{i_1 j_1}$, $g_{i_2 j_2}$ from the set $\{g_{11}, g_{12}, g_{21}, g_{22}\}$, see below, there are $K = 6$ different Jacobian matrices (20).

3.2.1 Motivation: locating Hopf using a minimally augmented system in Cl_matcont.

The two g_{ij} in (19) are selected from four g_{ij} , the entries of the solution matrix

$\begin{bmatrix} v_1 & v_2 \\ g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$, $v_1, v_2 \in \mathbb{R}^n$, $g_{ij} \in \mathbb{R}$, to the $(n+2)$ -dimensional *bordered system*:

$$\begin{bmatrix} A^2 + kI_n & w_{1,bor} & w_{2,bor} \\ v_{1,bor}^\top & 1 & 0 \\ v_{2,bor}^\top & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \\ g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} 0_{n \times 1} & 0_{n \times 1} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (21)$$

where $v_{1,bor}, v_{2,bor} \in \mathbb{R}^n$ are close to an orthonormal basis of $\mathcal{N}\left((A^0)^2 + \kappa I_n\right)$, and $w_{1,bor}, w_{2,bor} \in \mathbb{R}^n$ are close to an orthonormal basis of $\mathcal{N}\left(\left((A^0)^2 + \kappa I_n\right)^\top\right)$ (which ensures that the matrix in (21) is nonsingular).

Here the entries of the Jacobian matrix (20) are computed as follows: $(g_{i_1 j_1})_u$ and $(g_{i_2 j_2})_u$ are computed as

$$(g_{i_1 j_1})_u = -w_{i_1}^\top (A^2)_u v_{j_1}, \quad (g_{i_2 j_2})_u = -w_{i_2}^\top (A^2)_u v_{j_2}, \quad (22)$$

with w_i obtained by solving

$$\begin{bmatrix} (A^2 + kI_n)^\top & v_{1,bor} & v_{2,bor} \\ w_{1,bor}^\top & 1 & 0 \\ w_{2,bor}^\top & 0 & 1 \end{bmatrix} \begin{bmatrix} w_1 & w_2 \\ h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} 0_{n \times 1} & 0_{n \times 1} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad (23)$$

$(g_{i_1 j_1})_\alpha$ and $(g_{i_2 j_2})_\alpha$ are computed as

$$(g_{i_1 j_1})_\alpha = -w_{i_1}^\top (A^2)_\alpha v_{j_1}, \quad (g_{i_2 j_2})_\alpha = -w_{i_2}^\top (A^2)_\alpha v_{j_2}; \quad (24)$$

$(g_{i_1 j_1})_\kappa$ and $(g_{i_2 j_2})_\kappa$ are computed as

$$(g_{i_1 j_1})_\kappa = -w_{i_1}^\top v_{j_1}, \quad (g_{i_2 j_2})_\kappa = -w_{i_2}^\top v_{j_2}. \quad (25)$$

The primary motivation for this paper is to extend and improve results summarized in the following Remark.

Remark 5 *The system (19) is not unique, since there are $K = 6$ possible choices of the set $\{i_1, j_1, i_2, j_2\}$. CL_MATCONT uses QR decomposition of A to construct a minimally augmented system for Hopf continuation [11, pp. 185,186]). This method guarantees that the Jacobian matrix of (19) is nonsingular at the starting point, and it is suitable for small systems only.*

3.2.2 Locating Hopf using a minimally augmented system for large systems.

We modify (21) by replacing $A(x(s))$ by its restriction $C(x(s))$ onto $\mathcal{R}(x(s))$. We solve linear systems with the Jacobian matrix (20), which has the bordered form (7), using Algorithm 1, where we use Algorithm 2 to select the set $\{i_1, j_1, i_2, j_2\}$ so that the matrix \bar{D} in (9) has the smallest condition number.

The two g_{ij} in (19) are selected from four g_{ij} , the entries of the solution matrix $\begin{bmatrix} v_1 & v_2 \\ g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}$, $v_1, v_2 \in \mathbb{R}^m$, $g_{ij} \in \mathbb{R}$, to the $(m+2)$ -dimensional bordered system:

$$\begin{bmatrix} C^2 + \kappa I_m & w_{1,bor} & w_{2,bor} \\ v_{1,bor}^T & 1 & 0 \\ v_{2,bor}^T & 0 & 1 \end{bmatrix} \begin{bmatrix} v_1 & v_2 \\ g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} 0_{m \times 1} & 0_{m \times 1} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad (26)$$

where $v_{1,bor}, v_{2,bor} \in \mathbb{R}^m$ are close to an orthonormal basis of $\mathcal{N}\left((C^0)^2 + \kappa I_m\right)$, and $w_{1,bor}, w_{2,bor} \in \mathbb{R}^m$ are close to an orthonormal basis of $\mathcal{N}\left(\left((C^0)^2 + \kappa I_m\right)^T\right)$ (which ensures that the matrix in (26) is nonsingular). For $g = 0$, system (26) implies

$$\left[C^2 + \kappa I_m v \right] \begin{bmatrix} v_1 & v_2 \end{bmatrix} = 0_{m \times 2}, \quad \begin{bmatrix} v_{1,bor} & v_{2,bor} \end{bmatrix}^T \begin{bmatrix} v_1 & v_2 \end{bmatrix} = I_2.$$

Thus (19) and (26) hold at $x = x_0$, which is a regular zero of (19).

Setting up the Jacobian matrix (20). Similarly to (22), $(g_{i_1 j_1})_u$ and $(g_{i_2 j_2})_u$ are computed as

$$(g_{i_1 j_1})_u = -w_{i_1}^T (C^2)_u v_{j_1}, \quad (g_{i_2 j_2})_u = -w_{i_2}^T (C^2)_u v_{j_2}, \quad (27)$$

with w_i , analogously to (23), obtained by solving

$$\begin{bmatrix} (C^2 + \kappa I_m)^T & V_{bor} \\ W_{bor}^T & 0_{2 \times 2} \end{bmatrix} \begin{bmatrix} w_1 & w_2 \\ h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} 0_{m \times 2} \\ I_2 \end{bmatrix}, \quad (28)$$

where $(C^2)_u v_i$ is computed as

$$\begin{aligned} (C(x)^2)_u v_i &\approx Q_1^T \left[\frac{f_u(u + \delta \frac{y}{\|y\|}, \alpha) - f_u(u, \alpha)}{\delta} \|y\| \right. \\ &\quad \left. + C(x) Q_1^T \frac{f_u(u + \delta \frac{z}{\|z\|}, \alpha) - f_u(u, \alpha)}{\delta} \|z\| \right], \\ z &\equiv Q_1 v_i, \quad y \equiv Q_1 C(x) v_i \in \mathbb{R}^n. \end{aligned} \quad (29)$$

Similarly to (24), $(g_{i_1 j_1})_\alpha$ and $(g_{i_2 j_2})_\alpha$ are computed as

$$(g_{i_1 j_1})_\alpha = -w_{i_1}^T (C^2)_\alpha v_{j_1}, \quad (g_{i_2 j_2})_\alpha = -w_{i_2}^T (C^2)_\alpha v_{j_2}, \quad (30)$$

where $(C^2)_\alpha v_i$ is computed as

$$\begin{aligned} (C(x)^2)_\alpha v_i &\approx Q_1^T \left[\frac{f_u(u, \alpha + \delta) - f_u(u, \alpha)}{\delta} y \right. \\ &\quad \left. + C(x) Q_1^T \frac{f_u(u, \alpha + \delta) - f_u(u, \alpha)}{\delta} z \right], \\ z &\equiv Q_1 v_i, \quad y \equiv Q_1 C(x) v_i \in \mathbb{R}^n. \end{aligned} \quad (31)$$

Finally, $(g_{i_1 j_1})_\kappa$ and $(g_{i_2 j_2})_\kappa$ are computed in the same way as in (25):

$$(g_{i_1 j_1})_\kappa = -w_{i_1}^T v_{j_1}, \quad (g_{i_2 j_2})_\kappa = -w_{i_2}^T v_{j_2}. \quad (32)$$

Algorithm 3 *One step of the Newton's method for locating Hopf using the system (19) with the Jacobian matrices (20).*

Input: *Initial point $x = (u, \alpha)$, κ , the matrices f_x , C , and the matrices $V_{bor} = \begin{bmatrix} v_{1,bor} & v_{2,bor} \end{bmatrix}$, $W_{bor} = \begin{bmatrix} w_{1,bor} & w_{2,bor} \end{bmatrix}$.*

Step 1 *Solve $\begin{bmatrix} C^2 + kI_m & W_{bor} \\ V_{bor}^T & 0_{2 \times 2} \end{bmatrix} \begin{bmatrix} v_1 & v_2 \\ g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} 0_{m \times 2} \\ I_2 \end{bmatrix}$.*

Step 2 *Denote by M_i , $1 \leq i \leq K = 6$, the Jacobian matrices (20). Note, that M_i has the bordered form (7). Use Algorithm 2 to select M_i and compute the corresponding \bar{D}_i*

Step 3 *Solve $M_i \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta \kappa \end{bmatrix} = - \begin{bmatrix} f \\ g_{i_1 j_1} \\ g_{i_2 j_2} \end{bmatrix}$ for $(\Delta u, \Delta \alpha, \Delta \kappa) \in \mathbb{R}^{n+2}$ using Steps*

3 and 4 of the BED algorithm 1. Then set $\begin{bmatrix} u_{new} \\ \alpha_{new} \\ \kappa_{new} \end{bmatrix} = \begin{bmatrix} u \\ \alpha \\ \kappa \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta \kappa \end{bmatrix}$.

Step 4 *Compute $f_x = f_x(x_{new})$ and then $C = C(x_{new})$ using the CIS algorithm.*

Output: $x_{new} = (u_{new}, \alpha_{new})$, κ_{new} , f_x , C .

4 Torus Continuation

The minimally augmented system for the Torus continuation, (see [13], [17]) consists of $2n + 3$ scalar equations for $n + 4$ components $(u, T, \alpha, \kappa) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}$, $(i_1, j_1, i_2, j_2) \in \{1, 2\}$.

$$\begin{cases} \frac{du}{dt} - Tf(u, \alpha) = 0, \\ u(0) - u(1) = 0, \\ \int_0^1 \langle u(t), \dot{u}_{old}(t) \rangle dt = 0, \\ G_{i_1 j_1}(u, T, \alpha) = 0, \\ G_{i_2 j_2}(u, T, \alpha) = 0, \end{cases} \quad (33)$$

where $G_{i_1 j_1}$ and $G_{i_2 j_2}$ are selected from four G_{ij} , the entries of the solution matrix $\begin{bmatrix} v_1 & v_2 \\ G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}$, $v_1, v_2 \in \mathbb{R}^n$, $G_{ij} \in \mathbb{R}$, to the $(n+2)$ -dimensional *bordered system*:

$$N_3 \begin{bmatrix} v_1 & v_2 \\ G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} = \begin{bmatrix} 0_{n \times 2} \\ I_2 \end{bmatrix}. \quad (34)$$

Here

$$N_3 = \begin{bmatrix} D - Tf_u(u, \alpha) & w_{11} & w_{12} \\ \delta_0 - 2\kappa\delta_1 + \delta_2 & w_{21} & w_{22} \\ Int_{v_{01}} & 0 & 0 \\ Int_{v_{02}} & 0 & 0 \end{bmatrix} \quad (35)$$

where D is the time differentiation operator acting from $C^1([0, 1], \mathbb{R}^n)$ to $C^0([0, 1], \mathbb{R}^n)$; the bordering functions $v_{01}, v_{02}, w_{11}, w_{12}$, vectors w_{21} and w_{22} are chosen so that N_3 is nonsingular; $\delta_0, \delta_1, \delta_2$ are the Dirac operators defined by $\delta_i(f) = f(i)$ for $f \in C^1([0, 1], \mathbb{R}^n)$. $Int_{v_{0i}}$ is the operator defined by $Int_{v_{0i}} f = \int_0^1 v_{0i}^T(t) f(t) dt$ for $f \in C^1([0, 1], \mathbb{R}^n)$.

The system (33) is discretized by *orthogonal collocation* (see [13], [17]). Let N be the number of subintervals and m be the order of the interpolating polynomial at each subinterval. Then we have a discretized system of $n(Nm + 1) + 3$ equations for $n(Nm + 1) + 4$ components $(u_d, T, \alpha, \kappa) \in \mathbb{R}^{n(Nm+1)} \times \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}$:

$$\begin{cases} F(u_d, T, \alpha) = 0, \\ G_{i_1 j_1}(u_d, T, \alpha) = 0, \\ G_{i_2 j_2}(u_d, T, \alpha) = 0. \end{cases} \quad (36)$$

From (34) we get four equations $G_{ij} = 0, ((i, j) \in \{1, 2\})$ in case of Torus bifurcation. But we use Algorithm 2 to select the set $\{i_1, j_1, i_2, j_2\}$ so that the matrix \bar{D} in (9) has the smallest condition number. In this case, $k = 3$ and $K = 6$.

And we use Algorithm 1 to solve the linear system. In this case, $k = 3$.

n	Minimally augmented system	Standard augmented system
16, 384	6.7 s	10.2 s
32, 768	18.9 s	20.7 s
	3 Newton iterations	4 Newton iterations

Table 1: The 1-D Brusselator problem (locating Hopf point)

5 Examples.

All computations below are done using MATLAB 7.0 on a 3.0 GHz Pentium 4 PC.

Example 1 *The 1D Brusselator [23] is a well known model system for autocatalytic chemical reactions with diffusion:*

$$\begin{aligned} \frac{d_1}{l^2} u'' - (b+1)u + u^2v + a &= 0, & \frac{d_2}{l^2} v'' + bu - u^2v &= 0, & \text{in } \Omega = (0, 1), \\ u(0) = u(1) &= a, & v(0) = v(1) &= \frac{b}{a}. \end{aligned} \quad (37)$$

This problem exhibits a rich bifurcation scenario and has been used in the literature as a standard model for bifurcation analysis [25, 20, 9, 2, 8, 24]. We utilize the second order central difference discretization with uniform grid of N grid points. Since there are two unknowns per grid point, the dimension of the resulting discrete problem, which can be written in the form (2), is $n = 2N$. This discretization of the Brusselator is used in a CL_MATCONT example [13]. A constant solution to (2) is continued in parameter b , and a Hopf point is located for $b = 17.2056$. Table 1 shows a comparison between the amount of work required to locate the Hopf point using a minimally augmented system and a standard augmented system. As we can see, a standard augmented system takes more time than a minimally augmented system.

Example 2 *The 2D Brusselator*

$$\begin{aligned} \frac{d_1}{l^2} \Delta u - (b+1)u + u^2v + a &= 0, & \frac{d_2}{l^2} \Delta v + bu - u^2v &= 0, & \text{in } \Omega = (0, 1) \times (0, 1), \\ u|_{\partial\Omega} &= a, & v|_{\partial\Omega} &= \frac{b}{a}. \end{aligned} \quad (38)$$

We utilize the second order central difference discretization with uniform grid of N grid points in each of two directions. The dimension of the resulting discrete problem, which can be written in the form (2), is $n = 2N^2$. A constant solution to (2) is continued in b , and a Hopf point is located for $b = 22.0926$. Table 2 shows a comparison between the amount of work required to locate the Hopf point using a minimally augmented system and a standard augmented system. As we can see, a standard augmented system takes more time than a minimally augmented system.

Example 3 *The 1D Brusselator. The Hopf branch is continued in two parameters a and b starting at $(a, b) = (1.00000, 17.2056)$ and terminating at $(a, b) = (1.00228, 17.2443)$. See Table 3.*

n	Minimally augmented system	Standard augmented system
5,000	14.9 s	22.5 s
20,000	91.2 s 3 Newton iterations	134.1 s 4 Newton iterations

Table 2: The 2-D Brusselator problem (locating Hopf point)

n	total points	total time
32,768	12	414.7 s

Table 3: The 1-D Brusselator problem (continuation of the Hopf branch)

Remark 6 *The effect of choice of g_{ij} is illustrated for 1D Brusselator with $n = 512$. If we choose $(g_{i_1 j_1}, g_{i_2 j_2}) = (g_{11}, g_{22})$ to construct the system (19), the condition number of the Jacobian matrix (20) at the starting point is over 2×10^{22} , so no continuation is possible. However, if we choose g_{ij} using Algorithm 2, we get $(g_{i_1 j_1}, g_{i_2 j_2}) = (g_{11}, g_{21})$. In this case the condition number of (20) at the starting point is 6.9×10^5 , and Newton iteration converges.*

References

- [1] E. ALLGOWER AND K. GEORG, *Numerical path following*, in Handbook of Numerical Analysis, volume 5, P. G. Ciarlet and J. L. Lions, eds., North-Holland, 1997, pp. 3–207.
- [2] P. ASHWIN AND Z. MEI, *A Hopf bifurcation with Robin boundary conditions*, J. Dynamics and Differential Equations, 6 (1994), pp. 487–503.
- [3] W.-J. BEYN, A. CHAMPNEYS, E. J. DOEDEL, YU.A. KUZNETSOV, B. SANDSTED, 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.
- [4] 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.
- [5] 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.
- [6] ———, *Continuation of invariant subspaces for large bifurcation problems*, Tech. Report UCB/EECS-2006-13, EECS Department, University of California, Berkeley, February 13 2006.

- [7] D. BINDEL, W. DEMMEL, M. FRIEDMAN, W. GOVAERTS, AND Y. KUZNETSOV, *Bifurcation analysis of large equilibrium systems in matlab*, in Proceedings of the ICCS conference 2005, volume 3514/2005, Atlanta, GA, 2005, pp. 50–57.
- [8] C. S. CHIEN, Z. MEI, AND C. L. SHEN, *Numerical continuation at double bifurcation points of a reaction-diffusion problem*, Int. J. Bifur. and Chaos, 8 (1997), pp. 117–139.
- [9] G. DANGELMAYR, *Degenerate bifurcations near a double eigenvalue in the Brusselator*, J. Austral. Math. Soc. Ser. B, 28 (1987), pp. 486–535.
- [10] J. W. DEMMEL, L. DIECI, AND M. J. FRIEDMAN, *Computing connecting orbits via an improved algorithm for continuing invariant subspaces*, SIAM J. Sci. Comput., 22 (2001), pp. 81–94.
- [11] A. DHOOGHE, *MatCont: MATLAB Software for Bifurcations of Dynamical Systems*, PhD thesis, May 2005.
- [12] A. DHOOGHE, W. GOVAERTS, AND YU.A. KUZNETSOV, *MATCONT: A MATLAB package for numerical bifurcation analysis of odes.*, ACM TOMS., 29 (2003), pp. 141–164.
- [13] A. DHOOGHE, W. GOVAERTS, YU.A. KUZNETSOV, W. MESTROM, AND A. M. RIET, *MATLAB continuation software package CL_MATCONT*, Jan. 2003. <http://www.math.uu.nl/people/kuznet/cm/>.
- [14] L. DIECI AND M. J. FRIEDMAN, *Continuation of invariant subspaces*, Numerical Linear Algebra and Appl., 8 (2001), pp. 317–327.
- [15] E. DOEDEL, H. B. KELLER, AND J.-P. KERNÉVEZ, *Numerical analysis and control of bifurcation problems. I. Bifurcation in finite dimensions*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 1 (1991), pp. 493–520.
- [16] ———, *Numerical analysis and control of bifurcation problems. II. Bifurcation in infinite dimensions*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 1 (1991), pp. 745–772.
- [17] E. J. DOEDEL, W. GOVAERTS, AND Y. A. KUZNETSOV, *Computation of periodic solution bifurcations in odes using bordered systems*, SIAM J. Numer. Anal., 41 (2003), pp. 401–435.
- [18] M. FRIEDMAN, W. GOVAERTS, Y. KUZNETSOV, AND B. SAUTOIS, *Continuation of homoclinic orbits in matlab*, in Proceedings of the ICCS conference 2005, volume 3514/2005, Atlanta, GA, 2005, pp. 263–270.
- [19] M. J. 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.

- [20] M. GOLUBITSKY AND D. G. SCHAEFFER, *Singularities and groups in bifurcation theory, Vol 1*, Applied Mathematical Sciences 51. Springer-Verlag, New York, 1985.
- [21] W. GOVAERTS, *Numerical methods for bifurcations of dynamical equilibria*, SIAM, Philadelphia, 2000.
- [22] YU. A. KUZNETSOV, *Elements of Applied Bifurcation Theory, Third edition*, Springer-Verlag, New York, 2004.
- [23] R. LEFEVER AND I. PRIGOGINE, *Symmetry-breaking instabilities in dissipative systems II*, J. Chem. Phys., 48 (1968), pp. 1695–1700.
- [24] Z. MEI, *Numerical bifurcation analysis for reaction-diffusion equations*, PhD thesis, University of Marburg, 1997.
- [25] 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.