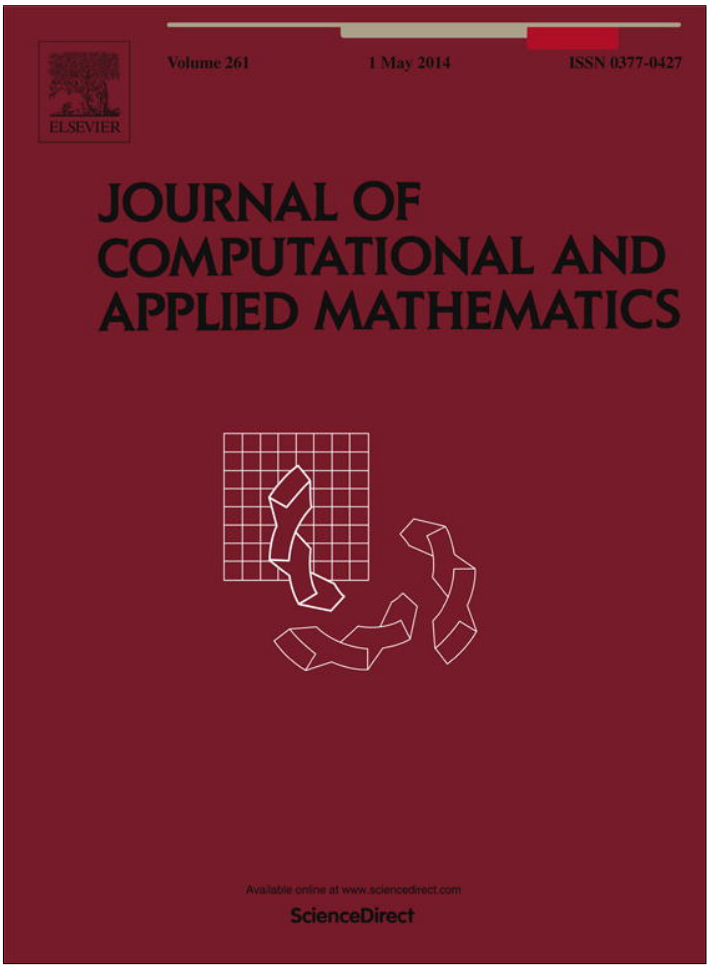

Journal of Computational and Applied Mathematics 261 (2014) 232–248

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

# Numerical computation of bifurcations in large equilibrium systems in MATLAB

D. Bindel [a], M. Friedman [b,*], W. Govaerts [c], J. Hughes [b], Yu.A. Kuznetsov [d]

[a] Department of Computer Science, Cornell University, Ithaca, NY 14853, United States
[b] Mathematical Sciences Department, University of Alabama in Huntsville, Huntsville, AL 35899, United States
[c] Department of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281-S9, B-9000 Ghent, Belgium
[d] Mathematisch Instituut Budapestlaan 6, 3584CD Utrecht, The Netherlands

ARTICLE INFO

*Article history:*
Received 6 December 2012
Received in revised form 7 October 2013

*Keywords:*
Bifurcations
Numerical continuation
Large systems
MATLAB

ABSTRACT

The Continuation of Invariant Subspaces (*CIS*) algorithm produces a smoothly-varying basis for an invariant subspace $\mathcal{R}(s)$ of a parameter-dependent matrix $A(s)$. We have incorporated the CIS algorithm into CL_MATCONT, a MATLAB package for the study of dynamical systems and their bifurcations. Using subspace reduction, we extend the functionality of CL_MATCONT to large-scale computations of bifurcations of equilibria. In this paper, we describe the algorithms and functionality of the resulting MATLAB bifurcation package CL_MATCONTL. The novel features include: new CIS-based, continuous, well-scaled test functions for codimension 1 and 2 bifurcations; detailed description of locators for large problems; and examples of bifurcation analysis in large sparse problems.



## 1. Introduction

Parameter-dependent Jacobian matrices provide important information about dynamical systems

$$\frac{du}{dt} = f(u,\alpha), \quad \text{where } u \in \mathbb{R}^n,\ \alpha \in \mathbb{R},\ f(u,\alpha) \in \mathbb{R}^n. \tag{1}$$

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

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

we look at the linearization $f_u(u(s),\alpha(s))$. For general background on dynamical systems theory we refer to the literature, in particular [1]. We are interested in continuation and bifurcation analysis of the stationary problem (2) in the case when the dimension $n$ is large. In this setting, direct methods to detect and locate bifurcations are expensive; but if we can multiply by $f_u$ quickly, we can use projection methods. For example, consider a spatial discretization of an elliptic partial differential equation. In this case, $f_u$ is typically large and sparse, but a small invariant subspace $\mathcal{R}(s)$ corresponding to a few eigenvalues near the imaginary axis provides information about stability and bifurcations.

Numerical continuation for large nonlinear systems of this form 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 that use a time integrator as a black box to identify the low-dimensional

* Corresponding author. Tel.: +1 2567229747; fax: +1 256 824 6173.
*E-mail addresses:* bindel@cs.cornell.edu (D. Bindel), friedman@math.uah.edu (M. Friedman), Willy.Govaerts@UGent.be (W. Govaerts), Hughes@email.uah.edu (J. Hughes), kuznet@math.uu.nl (Yu.A. Kuznetsov).

0377-0427/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cam.2013.10.034

invariant subspace where interesting dynamics take place; see e.g. [2–5] and references there. The review article [6] provides an overview of current techniques and typical applications of numerical bifurcation analysis in fluid dynamical problems.

To our knowledge, the C++ package LOCA [7] is the only general-purpose bifurcation analysis package that targets large-scale equilibrium systems on parallel computers. In 1-parameter continuation in LOCA, bifurcations are detected by finding an eigenvalue whose real part changes sign. Then augmented systems are used to locate fold, pitchfork, and Hopf bifurcations, and to continue these bifurcations with respect to a second parameter. PDE2PATH [8] is a recent MATLAB package for 1-parameter continuation and bifurcation in 2D elliptic systems. A simple bifurcation point is detected using sign $(\det f_u)$ as a test function, a bisection method is then used to locate it, and branch switching is also supported. We are not aware of any bifurcation packages for large systems that compute codimension-2 bifurcations on 1-parameter curves of bifurcations. The starting point for our work is the command line code CL_MATCONT [9]. CL_MATCONT and its GUI version MATCONT [10] are MATLAB packages for the study of small and moderate-size dynamical systems and their bifurcations. They both use *minimally augmented systems* [11] for continuation of bifurcations.

The Continuation of Invariant Subspaces *(CIS) algorithm* produces a smooth orthonormal basis for an invariant subspace $\mathcal{R}(s)$ of a parameter-dependent matrix $A(s)$ [12–16]. The CIS algorithm uses projection methods to deal with large problems, though other approaches have been used in closely related work [17]. We have incorporated the CIS algorithm into CL_MATCONTL to extend its functionality to large scale bifurcation computations of equilibria via subspace reduction. The result is a MATLAB bifurcation package CL_MATCONTL [18,19]. In this paper we describe the algorithms and functionality of CL_MATCONTL and give representative examples. Some initial results in this direction are reported in [20,21]. See also [16,22], and references there for analysis of the algorithms in CL_MATCONTL. See [23] for an application of CL_MATCONTL to a bifurcation analysis of the mitochondrial respiratory chain.

The novel features include: new CIS-based, continuous, well-scaled test functions for codimension 1 and 2 bifurcations, especially useful for large problems; detailed description of locators for large problems; and examples of bifurcation analysis of large, sparse equilibrium systems with 20,000–25,000 unknowns, including detecting and locating codimension 1 and 2 bifurcations.

The rest of the paper is organized as follows. Section 2 contains preliminaries. In Section 3, we introduce and analyze four new CIS-based test functions for detecting fold, Hopf, and branch point bifurcations. This may lead to a more reliable detection of singularities. We also note that the CIS algorithm ensures that only eigenvalues of a matrix that is much smaller than $A(s)$, namely, the restriction $C(s) := A(s)|_{\mathcal{R}(s)}$, can cross the imaginary axis, so that $C(s)$ provides most of the relevant information about bifurcations. In addition, the subspace $\mathcal{R}(s)$ that is continued is adapted to track behavior relevant to bifurcations. In Section 4, we use subspace reduction to extend to large-scale equilibrium problems a minimally augmented system technique for locating fold, Hopf, and branch point bifurcations, and for branch switching. Some material in this section is a review of previous results (to make this paper self-contained), as indicated in the text. In Section 5, we extend the results in Sections 3 and 4 to continuation of fold and Hopf bifurcations and to detecting and locating most generic codimension-2 bifurcations on those two curves. Section 6 contains several representative examples of bifurcation analysis in large sparse equilibrium problems (2). These examples support the assertion that the new CIS-based algorithms can accurately, reliably, and within a reasonable time detect, locate, and continue singularities of interest in large systems.

## 2. Preliminaries

*Notation.* If $A$ is a matrix, then $A^*$ denotes the transposed matrix in the real case and the complex conjugate transposed matrix in the complex case.

Let $f(x) \equiv f(u, \alpha)$ be a smooth function and $x_0 = (u_0, \alpha_0)$ a given point. We then use the notation

$$f_u^0 v = f_u(x_0)v, \qquad f_{uu}^0 vw = f_{uu}(x_0)[v, w],$$

as a short hand for the derivatives of $f$ with respect to $u$ evaluated at $x_0$ and as a multi-linear form applied to vectors $v$ and $w$.

### 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. For some $m \ll n$, let

$$\Lambda_1(s) := \{\lambda_i(s)\}_{i=1}^m, \qquad \operatorname{Re}\lambda_m \le \cdots \le \operatorname{Re}\lambda_{m_u+1} \le 0 < \operatorname{Re}\lambda_{m_u} \le \cdots \le \operatorname{Re}\lambda_1, \tag{3}$$

be a small set consisting of the rightmost eigenvalues of $A(s)$. Then an application of the CIS algorithm [15,16] to $A(s)$ produces an orthonormal basis $Q_1(s) \in \mathbb{R}^{n\times m}$ for the invariant subspace $\mathcal{R}(s)$ corresponding to $\Lambda_1(s)$, and we consider the restriction

$$C(s) := T_{11}(s) = Q_1^*(s)A(s)Q_1(s) \in \mathbb{R}^{m\times m}, \tag{4}$$

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

$$A(s) = \begin{bmatrix} Q_1(s) & Q_2(s) \end{bmatrix} \begin{bmatrix} T_{11}(s) & T_{12}(s) \\ 0 & T_{22}(s) \end{bmatrix} \begin{bmatrix} Q_1(s) & Q_2(s) \end{bmatrix}^*. \tag{5}$$

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

Consider linear systems of the form

$$M\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}, \tag{6}$$

where $M$ has the bordered form

$$M = \begin{bmatrix} A & B \\ C^* & D \end{bmatrix}, \tag{7}$$

with $A \in \mathbb{R}^{n\times n}$ 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 $k \ll n$; see [24,25], and [11, Section 3.6]. When both $M$ and $A$ are well-conditioned, (6) can be solved accurately by two different block $LU$ factorizations of $M$: BED (block elimination Doolittle), based on the factorization

$$\begin{bmatrix} A & B \\ C^* & D \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ W^* & I_k \end{bmatrix}\begin{bmatrix} A & B \\ 0 & \Delta^* \end{bmatrix}; \tag{8}$$

and BEC (block elimination Crout), based on the factorization

$$\begin{bmatrix} A & B \\ C^* & D \end{bmatrix} = \begin{bmatrix} A & 0 \\ C^* & \Delta \end{bmatrix}\begin{bmatrix} I_n & V \\ 0 & I_k \end{bmatrix}. \tag{9}$$

When $M$ is well-conditioned and $A$ is ill-conditioned, (6) can be solved accurately by a combination of BED and BEC, called BEM (mixed block elimination) in the case $k = 1$ and BEMW in the case $k > 1$.

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

*Step* 1 Solve

$$A^*\bar{C} = C.$$

*Step* 2 Compute

$$\bar{D} \equiv D - \bar{C}^*B.$$

*Step* 3 Solve

$$\bar{D}y = g - \bar{C}^*f. \tag{10}$$

*Step* 4 Solve.

$$Ax = f - By.$$

**Remark 1.** We use the SVD decomposition, which has better stability properties than $LU$ factorization, to solve the small linear system (10).

**Remark 2.** We use the BEM algorithm to solve linear systems for locating and continuing folds since $A$ can be ill-conditioned in this case. We use the BED algorithm to solve linear systems for locating and continuing Hopf bifurcations. In this case $A$ 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: $M_i := \begin{bmatrix} A & B \\ C_i^* & D_i \end{bmatrix}$ 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 $1 \le i \le K$.

Step 1 Solve the linear system:

$$A^*\begin{bmatrix} \bar{C}_1 & \cdots & \bar{C}_K \end{bmatrix} = \begin{bmatrix} C_1 & \cdots & C_K \end{bmatrix}. \tag{11}$$

Step 2 Compute

$$\begin{bmatrix} \bar{D}_1 & \cdots & \bar{D}_K \end{bmatrix} = \begin{bmatrix} D_1 & \cdots & D_K \end{bmatrix} - \begin{bmatrix} \bar{C}_1^* & \cdots & \bar{C}_K^* \end{bmatrix} B. \tag{12}$$

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

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

**Remark 3.** In 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 a Hopf bifurcation, and $k = 3$ and $K = 6$ in the case of a continuation of a Hopf bifurcation (the size of the border is increased by one). In both cases, these computations are cheap.

## 3. Detecting fold, Hopf, and branch point bifurcations

### 3.1. Introduction

*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 system (2). We write the Jacobian matrix along this path as $A(x(s)) = f_u(x(s))$. We will also use the notation $A = A(x)$, $A^0 = A(x_0)$, $f_x = f_x(x)$, and, for restriction $C(x)$ of $A(x)$ onto $\mathcal{R}(s)$ defined by (4), $C = C(x)$, $C^0 = C(x_0)$.

A solution point $x_0 = x(s_0)$ is a *bifurcation point* if $\mathrm{Re}\,\lambda_i(s_0) = 0$ for at least one eigenvalue $\lambda_i(s_0)$ of $A(s_0)$. A *test function* $\phi(s) = \psi(x(s))$ 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

$$\psi\,(x(s_k))\,\psi\,(x(s_{k+1})) < 0. \tag{13}$$

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

$$\begin{cases} f(x) = 0, \\ g(x) = 0, \end{cases} \tag{14}$$

for an appropriate function $g$.

On a solution branch $x(s)$ of the system (2), we consider the two generic codimension-1 bifurcations, *fold* or *limit point* (LP) and *Hopf* (H) bifurcations, and also *branch point* (BP) bifurcations.

**Definition 1.** We call $x_0$ a *simple fold* if

(S1) $\frac{1}{2} w^* f^0_{uu} v v \neq 0$, and
(S2) $w^* f^0_\alpha \neq 0$,

where $A^0 v = A^{0*} w = 0$, $v^* v = w^* v = 1$.

**Definition 2.** We call $x_0$ a *simple Hopf point* if $A^0$ has an imaginary conjugate eigenpair $\lambda = \pm i\omega$, $\omega > 0$, and $d\,(\mathrm{Re}\,\lambda)\,/ds \neq 0$ at $s_0$.

*Test functions for bifurcations in* CL_MATCONT. To detect and locate branch points, Hopf points, and limit points, CL_MATCONT uses the following test functions (see e.g. [1,11,26,9]):

$$\psi^M_{BP}\,(x(s)) := \det \begin{bmatrix} A & f_\alpha \\ \dot{u}^* & \dot{\alpha} \end{bmatrix}, \tag{15}$$

$$\psi^M_H\,(x(s)) := \det\,[2A(s) \odot I_n] = \prod_{1 \le i < j \le n} \left(\lambda_i(s) + \lambda_j(s)\right), \tag{16}$$

$$\psi^M_{LP}\,(x(s)) := \dot{\alpha}, \tag{17}$$

where $\dot{x} \equiv dx/ds$ and $\odot$ is the bialternate product [11], $2A(s) \odot I_n \in \mathbb{R}^{b(n)\times b(n)}$, $b(n) = \frac{n(n-1)}{2}$. The bifurcations are defined by:

$$\mathrm{BP} : \psi^M_{BP} = 0, \qquad \mathrm{H} : \psi^M_H = 0, \qquad \mathrm{LP} : \psi^M_{LP} = 0, \qquad \psi^M_{BP} \neq 0. \tag{18}$$

### 3.2. Numerical continuation of equilibria

For numerical continuation of equilibria, one solves (2) together with an additional scalar equation. To use a Newton-like method, one needs the Jacobian matrix of the resulting system, which has a bordered form (7). Here we can multiply by $A$ quickly, both $M$ and $A$ are well-conditioned, and $k = 1$. Hence we use BED to solve (6) in this case.

To compute the tangent vector $v_0$ at the starting point $x_0 = (u_0, \alpha_0)$, we use the equation $f^0_u \dot{u} + f^0_\alpha \dot{\alpha} = 0$. Specifically, we first solve the linear system $A^0 v = f^0_\alpha$ for $v$, and then set $v_0 = \begin{bmatrix} v & -1 \end{bmatrix}^*$, $v_0 = \frac{v_0}{\|v_0\|}$.

### 3.3. Detecting singularities

To detect fold points, branch points, and Hopf points, we define four new continuous, well-scaled test functions as follows.

**Lemma 3.** *Let* $\Lambda(s) := \{\lambda_i(s)\}_{i=1}^n$ *be the set of eigenvalues of* $A(s)$*, and let* $\Lambda_1(s) = \{\lambda_i(s)\}_{i=1}^m$ *be a given set* (3) *of the rightmost eigenvalues of* $A(s)$*, where at least* $\mathrm{Re}\,(\lambda_m(s)) < 0$*. Define also*

$$m_u(s) := \mathrm{card}\{\lambda_i(s) : 1 \le i \le m, \mathrm{Re}(\lambda_i(s)) > 0\} \tag{19}$$

*and a piecewise smooth function*

$$\lambda_{\min}(s) := \min\{|(\lambda_i(s))| : 1 \le i \le m\}. \tag{20}$$

*Also given is the factorization*

$$\begin{bmatrix} A & f_\alpha \\ \dot{u}^* & \dot{\alpha} \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ w^* & 1 \end{bmatrix} \begin{bmatrix} A & b \\ 0 & \delta^* \end{bmatrix}. \tag{21}$$

*Further, define*

$$M_u(s) := \mathrm{card}\{\lambda_i(s) + \lambda_j(s) : 1 \le i < j \le m, \mathrm{Re}(\lambda_i(s) + \lambda_j(s)) > 0\} \tag{22}$$

*and a piecewise smooth function:*

$$\mu_{\min}(s) := \min\left\{|(\lambda_i(s) + \lambda_j(s))| : 1 \le i < j \le m\right\}. \tag{23}$$

*Then the new piecewise smooth CIS test functions defined below satisfy:*

1.
$$\psi_{BP}(x(s)) := \lambda_{\min}(s)(-1)^{m_u(s)}\mathrm{sign}(\delta^*(s)) \tag{24}$$
*changes sign if and only if* $\psi_{BP}^M(x(s))$ *does.*

2.
$$\psi_{LP}(x(s)) := \lambda_{\min}(s)(-1)^{m_u(s)} \tag{25}$$
*changes sign if and only if* $\psi_{LP}^M(x(s))$ *does and* $\psi_{BP}(x(s))$ *is nonzero.*

3.
$$\psi_H^{(1)}(x(s)) := \mu_{\min}(s)(-1)^{M_u(s)} \tag{26}$$
*changes sign if and only if* $\psi_H^M(x(s))$ *does. Moreover,*
$$\psi_H^{(2)}(x(s)) := (-1)^{m_u^{\mathrm{pairs}}}, \tag{27}$$
*where* $m_u^{\mathrm{pairs}}(s)$ *is the number of complex eigenvalues from* $\Sigma_1(s)$ *with* $\mathrm{Re}(\lambda_i(s)) \ge 0$ *and* $\mathrm{Im}(\lambda_i(s)) > 0$*, changes sign if and only if the number of unstable complex conjugate eigenpairs from* $\Lambda_1(s)$ *changes.*

**Proof.** Easy. ■

Using the above test functions, we can detect the singularities:

$$\text{LP}: \psi_{BP}(x(s_k))\,\psi_{BP}(x(s_{k+1})) > 0 \quad \text{and} \quad \psi_{LP}(x(s_k))\,\psi_{LP}(x(s_{k+1})) < 0, \tag{28}$$

$$\text{H}: \psi_H^{(1)}(x(s_k))\,\psi_H^{(1)}(x(s_{k+1})) < 0 \quad \text{and} \quad \psi_H^{(2)}(x(s_k))\,\psi_H^{(2)}(x(s_{k+1})) < 0, \tag{29}$$

$$\text{BP}: \psi_{BP}(x(s_k))\,\psi_{BP}(x(s_{k+1})) < 0. \tag{30}$$

**Remark 4.** The principal advantage of the new CIS-based test function $\psi_{LP}$ (25), as compared with the standard one $\psi_{LP}^M$ (15), is that $\psi_{LP}$ is based on an eigenvalue, like all other new CIS-based test functions in CL_MATCONTL. Hence it is scaled in the same way as all other new CIS-based test functions, whereas the $\alpha$-component of the normalized tangent vector to the curve might be very small.

**Remark 5.** The test function $\psi_H^{(1)}$ replaces $\psi_H^M$. Clearly, both $\psi_H^M$ and $\psi_H^{(1)}$ are zero not only when $A(s)$ has a pure imaginary pair of eigenvalues ($\pm i\omega$), but also if there is a pair of real eigenvalues with sum zero; $\psi_H^{(2)}$ is used to exclude this case.

## 4. Locators for fold, Hopf and branch points

We use *minimally augmented systems* (see [11,10,9]) with $A(x(s))$ replaced by its restriction $C(x(s))$ onto $\mathcal{R}(s)$ whenever possible.

### 4.1. Locator for a fold

Let $x_0$ be a fold point. Then $A^0$ has rank $n-1$. The minimally augmented system to locate $x_0$ consists of $n+1$ scalar equations for $n+1$ components $x = (u, \alpha) \in \mathbb{R}^n \times \mathbb{R}$,

$$\begin{cases} f(x) = 0, \\ g(x) = 0, \end{cases} \tag{31}$$

where $g = g(x)$ is the last component of the solution vector $(v, g) \in \mathbb{R}^m \times \mathbb{R}$ to the $(m+1)$-dimensional bordered system:

$$\begin{bmatrix} C & w_{\text{bor}} \\ v_{\text{bor}}^* & 0 \end{bmatrix} \begin{bmatrix} v \\ g \end{bmatrix} = \begin{bmatrix} 0_{m\times 1} \\ 1 \end{bmatrix}, \tag{32}$$

where $v_{\text{bor}} \in \mathbb{R}^m$ is close to a null vector of $C^0$, and $w_{\text{bor}} \in \mathbb{R}^m$ is close to a null vector of $C^{0*}$ (which ensures that the matrix in (32) is nonsingular). For $g = 0$, system (32) implies $Cv = 0$, $v_{\text{bor}}^* v = 1$. Thus (31) and (32) hold at $x = x_0$, which is a regular zero of (31).

The system (31) is solved using Newton's method, and its Jacobian matrix is:

$$J = \begin{bmatrix} f_x \\ g_x \end{bmatrix} = \begin{bmatrix} A & f_\alpha \\ g_u & g_\alpha \end{bmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}, \tag{33}$$

where $g_x$ is computed as

$$g_x = -w^* C_x v, \tag{34}$$

with $w$ obtained by solving

$$\begin{bmatrix} C^* & v_{\text{bor}} \\ w_{\text{bor}}^* & 0 \end{bmatrix} \begin{bmatrix} w \\ g \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{35}$$

To reduce the cost of computing $J$, we use an approximation

$$C_x v = \left(Q_1^*(x) A(x) Q_1(x)\right)_x v \approx Q_1^* A_x Q_1 v \tag{36}$$

and a first-order finite difference approximation

$$A_x(x)v = (f_x)_u v = \frac{f_x\left(u + \delta \frac{z}{\|z\|}, \alpha\right) - f_x(u,\alpha)}{\delta} \|v\| + O(\delta), \quad z := Q_1 v \in \mathbb{R}^n.$$

Combining the above two equations, we get

$$C_x(x)v \approx Q_1^* \frac{f_x\left(u + \delta \frac{z}{\|z\|}, \alpha\right) - f_x(u,\alpha)}{\delta} \|z\|, \quad z := Q_1 v \in \mathbb{R}^n. \tag{37}$$

Finally we note that at each Newton step for solving (31), linear systems with the matrix (33) should be solved by the BEM, since the matrix (33) has the form (7) with $k = 1$ and $A$ can be ill-conditioned.

**Remark 6.** An approximation (36) to $C_x(x)v$ is accurate only when the subspace $\mathcal{R}(s)$ does not vary much at $x$ : $(Q_1(x))_x$ is 'small'. This does not seem to cause difficulties in practice. In all our computations, we observed convergence of the Newton iteration, possibly with step size reduction in some cases.

**Algorithm 3.** One step of Newton's method for locating a fold.

Input: Initial point $x = (u, \alpha)$, the matrices $f_x$, $C$ and parameters $v_{\text{bor}}$, $w_{\text{bor}}$.

Step 1 Set $B := \begin{bmatrix} C & w_{\text{bor}} \\ v_{\text{bor}}^* & 0 \end{bmatrix}$.

Step 2 Solve $B \begin{bmatrix} v \\ g \end{bmatrix} = \begin{bmatrix} 0_{m\times 1} \\ 1 \end{bmatrix}$ for $(v, g) \in \mathbb{R}^m \times \mathbb{R}$.

Step 3 Solve $B^* \begin{bmatrix} w \\ g \end{bmatrix} = \begin{bmatrix} 0_{m\times 1} \\ 1 \end{bmatrix}$ for $(w, g) \in \mathbb{R}^m \times \mathbb{R}$.

Step 4 Set $v_{\text{bor,new}} = \frac{v}{\|v\|}$, $w_{\text{bor,new}} = \frac{w}{\|w\|}$.

Step 5 Compute $g_x = -(Q_1 w)^* \frac{f_x(u+\delta\frac{z}{\|z\|},\alpha) - f_x(u,\alpha)}{\delta} \|z\|$, $z = Q_1 v$.

Step 6 Set $J = \begin{bmatrix} f_x \\ g_x \end{bmatrix}$, compute $\begin{bmatrix} f \\ g \end{bmatrix}$.

Step 7 Solve $J\Delta x = -\begin{bmatrix} f \\ g \end{bmatrix}$ for $\Delta x \in \mathbb{R}^{n+1}$ and set $x_{\text{new}} = x + \Delta x$.

Step 8 Compute $f_x = f_x(x_{\text{new}})$ and then $C = C(x_{\text{new}})$ using the CIS algorithm.

Output: $x_{\text{new}} = (u_{\text{new}}, \alpha_{\text{new}})$.

Once the fold point $x_0 = (u_0, \alpha_0)$ is computed, the corresponding quadratic normal form coefficient

$$a := \frac{1}{2}\left(\widehat{w}, B\left(\widehat{v}, \widehat{v}\right)\right) \equiv \frac{1}{2} \widehat{w}^* f_{uu}^0 \widehat{v}\,\widehat{v} \tag{38}$$

is computed approximately as

$$a \approx a_h := \frac{1}{2\delta^2}\widehat{w}^*\left[f\left(u_0+\delta\widehat{v},\alpha_0\right)+f\left(u_0-\delta\widehat{v},\alpha_0\right)\right], \quad \widehat{v}\approx\frac{Q_1 v}{\|Q_1 v\|}, \tag{39}$$

where $\widehat{w}$ is computed directly from $A^*\widehat{w}=0$, $\widehat{v}^*\widehat{w}=1$.

### 4.2. Locator for a Hopf bifurcation

We summarize here results from [27]. Let $x_0$ be a Hopf point, and let $\kappa_0$ be the square of the imaginary part of the Hopf bifurcation eigenvalue ($\lambda = i\omega_0$). Then $A^0$ has rank $n$. The minimally augmented system to locate $x_0$ consists of $n+2$ scalar equations for $n+2$ components $(x,\kappa)=(u,\alpha,\kappa)\in\mathbb{R}^n\times\mathbb{R}\times\mathbb{R}$, $(i_1,j_1,i_2,j_2)\in\{1,2\}$:

$$\begin{cases} f(x)=0, \\ g_{i_1j_1}(x,\kappa)=0, \\ g_{i_2j_2}(x,\kappa)=0. \end{cases} \tag{40}$$

The Jacobian matrix of system (40) is:

$$\begin{bmatrix} f_u & f_\alpha & 0_{n\times 1} \\ (g_{i_1j_1})_u & (g_{i_1j_1})_\alpha & (g_{i_1j_1})_\kappa \\ (g_{i_2j_2})_u & (g_{i_2j_2})_\alpha & (g_{i_2j_2})_\kappa \end{bmatrix} \in \mathbb{R}^{(n+2)\times(n+2)}, \tag{41}$$

where $g_{ij}\equiv g_{ij}(x,\kappa)$. The two $g_{ij}$ in (40) 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*:

$$\begin{bmatrix} C^2+kI_m & w_{1,\mathrm{bor}} & w_{2,\mathrm{bor}} \\ v_{1,\mathrm{bor}}^* & 0 & 0 \\ v_{2,\mathrm{bor}}^* & 0 & 0 \end{bmatrix}\begin{bmatrix} v_1 & v_2 \\ g_{11} & g_{12} \\ g_{21} & g_{22}\end{bmatrix} = \begin{bmatrix} 0_{m\times 1} & 0_{m\times 1} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{42}$$

where $v_{1,\mathrm{bor}}, v_{2,\mathrm{bor}}\in\mathbb{R}^m$ are close to an orthonormal basis of $\mathcal{N}\left(\left(C^0\right)^2+\kappa I_m\right)$, and $w_{1,\mathrm{bor}}, w_{2,\mathrm{bor}}\in\mathbb{R}^m$ are close to an orthonormal basis of $\mathcal{N}\left(\left(\left(C^0\right)^2+\kappa I_m\right)^*\right)$ (which ensures that the matrix in (42) is nonsingular). For $g=0$, system (42) implies

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

Thus (40) and (42) hold at $(x,\kappa)=(x_0,\kappa_0)$, which is a regular zero of (40).

**Remark 7.** Depending on the possible choice of $i_1, j_1, i_2, j_2$, there are $K=6$ different Jacobian matrices (41). We solve linear systems with the Jacobian matrix (41), which has the bordered form (7), using Algorithm 1; and we use Algorithm 2 to select the set $\{i_1,j_1,i_2,j_2\}$ so that the matrix $\bar{D}$ in (10) has the smallest condition number.

*Setting up the Jacobian matrix* (41). The entries $(g_{i_1j_1})_u$ and $(g_{i_2j_2})_u$ are computed as

$$(g_{i_1j_1})_u=-w_{i_1}^*(C^2)_u v_{j_1}, \qquad (g_{i_2j_2})_u=-w_{i_2}^*(C^2)_u v_{j_2}, \tag{43}$$

with $w_i$ obtained by solving

$$\begin{bmatrix} (C^2+\kappa I_m)^* & v_{1,\mathrm{bor}} & v_{2,\mathrm{bor}} \\ w_{1,\mathrm{bor}}^* & 0 & 0 \\ w_{2,\mathrm{bor}}^* & 0 & 0 \end{bmatrix}\begin{bmatrix} w_1 & w_2 \\ h_{11} & h_{12} \\ h_{21} & h_{22}\end{bmatrix} = \begin{bmatrix} 0_{m\times 2} \\ I_2 \end{bmatrix}, \tag{44}$$

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

$$(C(x)^2)_u v_i \approx Q_1^*\left[\frac{f_u\left(u+\delta\frac{y}{\|y\|},\alpha\right)-f_u(u,\alpha)}{\delta}\|y\|+C(x)Q_1^*\frac{f_u\left(u+\delta\frac{z}{\|z\|},\alpha\right)-f_u(u,\alpha)}{\delta}\|z\|\right], \tag{45}$$

$z\equiv Q_1 v_i, \qquad y\equiv Q_1 C(x) v_i\in\mathbb{R}^n.$

Furthermore, $(g_{i_1j_1})_\alpha$ and $(g_{i_2j_2})_\alpha$ are computed as

$$(g_{i_1j_1})_\alpha=-w_{i_1}^*(C^2)_\alpha v_{j_1}, \qquad (g_{i_2j_2})_\alpha=-w_{i_2}^*(C^2)_\alpha v_{j_2}, \tag{46}$$

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

$$(C(x)^2)_\alpha v_i \approx Q_1^* \left[ \frac{f_u(u, \alpha+\delta) - f_u(u,\alpha)}{\delta} y + C(x) Q_1^* \frac{f_u(u,\alpha+\delta) - f_u(u,\alpha)}{\delta} z \right], \tag{47}$$
$$z \equiv Q_1 v_i, \qquad y \equiv Q_1 C(x) v_i \in \mathbb{R}^n.$$

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

$$(g_{i_1 j_1})_\kappa = -w_{i_1}^* v_{j_1}, \qquad (g_{i_2 j_2})_\kappa = -w_{i_2}^* v_{j_2}. \tag{48}$$

**Algorithm 4.** One step of Newton's method for locating a Hopf bifurcation using the system (40) with the Jacobian matrices (41).

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

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

Step 2 Denote by $M_i$, $1 \le i \le K = 6$, the Jacobian matrices (41). 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 $\begin{bmatrix} u_{\text{new}} \\ \alpha_{\text{new}} \\ \kappa_{\text{new}} \end{bmatrix} = \begin{bmatrix} u \\ \alpha \\ \kappa \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta\kappa \end{bmatrix}$.

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

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

Once the Hopf point $x_0 = (u_0, \alpha_0)$ is computed, the corresponding quadratic normal form coefficient (the first Lyapunov coefficient) $l_1$ is given by

$$l_1 := \frac{1}{2}\text{Re}\langle \widehat{w}, C(\widehat{v}, \widehat{v}, \overline{\widehat{v}}) + B(\widehat{v}, (2i\omega_0 - I_n)^{-1} B(\widehat{v}, \widehat{v})) - 2B(\widehat{v}, A^{-1} B(\widehat{v}, \overline{\widehat{v}}))\rangle, \tag{49}$$

where $A^0\widehat{v} = i\omega_0\widehat{v}$, $A^{0*}\widehat{w} = -i\omega_0\widehat{w}$, $\widehat{w}^*\widehat{w} = \widehat{v}^*\widehat{v} = 1$, and

$$B(p, q) = f_{uu}^0 pq, \qquad C(p, q, r) = f_{uuu}^0 pqr, \quad p, q, r \in \mathbb{R}^n.$$

Then $l_1$ is approximated as follows: $\widehat{v}$ is computed from $C(x_0)v = i\omega_0 v$, $\widehat{v} = Q_1 v$, and $\widehat{w}$ is computed as above, and

$$B(p, p) \approx \frac{1}{h^2}[f(x_0 + hp, \alpha_0) + f(x_0 - hp, \alpha_0)], \tag{50}$$
$$B(p, q) = \frac{1}{4}[B(p+q, p+q) - B(p-q, p-q)],$$
$$C(p, p, p) \approx \frac{1}{8h^3}[f(x_0 + 3hp, \alpha_0) - 3f(x_0 + hp, \alpha_0) + 3f(x_0 - hp, \alpha_0) - f(x_0 - 3hp, \alpha_0)],$$
$$C(p, p, q) = \frac{1}{6}\left(C(p+q, p+q, p+q) - C(p-q, p-q, p-q)\right) - \frac{1}{3}C(q, q, q).$$

### 4.3. Branching

Let $x_0 = x(s_0)$ be a *a simple singular point* (see e.g. [26]), i.e. $f_x^0 = f_x(x_0)$ has rank $n-1$ and

$$\mathcal{N}\left(f_x^0\right) = \text{Span}\left\{v_1^0, v_2^0\right\}, \qquad \mathcal{N}\left(\left(f_x^0\right)^*\right) = \text{Span}\left\{\psi^0\right\}.$$

#### 4.3.1. Locator for a branch point

We use a minimally augmented system [28–30] of $n+2$ scalar equations for $n+2$ components $(x, \mu) = (u, \alpha, \mu) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}$,

$$\begin{aligned} &f(x) + \mu w_{\text{bor}} = 0, \\ &g_1(x) = 0, \\ &g_2(x) = 0, \end{aligned} \tag{51}$$

where $\mu$ is an unfolding parameter, $w_{\text{bor}} \in \mathbb{R}^n$ is fixed, and $g_1 = g_1(x)$, $g_2 = g_2(x) \in \mathbb{R}$ are computed as the last row of the solution matrix $\begin{bmatrix} v_1 & v_2 \\ g_1 & g_2 \end{bmatrix}$, $v_1, v_2 \in \mathbb{R}^{n+1}$, to the $(n+2)$-dimensional bordered system

$$\begin{bmatrix} f_x(x) & w_{\text{bor}} \\ V_{\text{bor}}^* & 0_{2\times 1} \end{bmatrix} \begin{bmatrix} v_1 & v_2 \\ g_1 & g_2 \end{bmatrix} = \begin{bmatrix} 0_{n\times 2} \\ I_2 \end{bmatrix}, \qquad V_{\text{bor}} = \begin{bmatrix} v_{1,\text{bor}} & v_{2,\text{bor}} \end{bmatrix}, \tag{52}$$

where $v_{1,\text{bor}}, v_{2,\text{bor}} \in \mathbb{R}^{n+1}$ are close to an orthonormal basis of $\mathcal{N}\left(f_x^0\right)$, and $w_{\text{bor}}$ is close to the null vector of $\left(f_x^0\right)^*$.

The system (51) is solved by Newton's method [28] with the modifications in [29]. Newton's method is globalized by combining it with a bisection algorithm on the solution curve.

**Algorithm 5.** One step of Newton's method for locating a branch point.

Input: Initial point $x = (u, \alpha)$, the matrix $f_x = f_x(x)$, and the parameters $w_{\text{bor}}$ and $V_{\text{bor}} = \begin{bmatrix} v_{1,\text{bor}} & v_{2,\text{bor}} \end{bmatrix}$.

Step 1 Set $B := \begin{bmatrix} f_x & w_{\text{bor}} \\ V_{\text{bor}}^* & 0_{2\times 1} \end{bmatrix}$.

Step 2 Solve $B \begin{bmatrix} v_1 & v_2 \\ g_1 & g_2 \end{bmatrix} = \begin{bmatrix} 0_{n\times 2} \\ I_2 \end{bmatrix}$ for $(v_1, g_1), (v_2, g_2) \in \mathbb{R}^{n+1} \times \mathbb{R}$.

Step 3 Solve $B \begin{bmatrix} \widehat{x} \\ \widehat{\mu} \end{bmatrix} = -\begin{bmatrix} f \\ 0_{2\times 1} \end{bmatrix}$ for $(\widehat{x}, \widehat{\mu}) \in \mathbb{R}^{n+1} \times \mathbb{R}$.

Step 4 Solve $B^* \begin{bmatrix} \psi \\ g \end{bmatrix} = \begin{bmatrix} 0_{(n+1)\times 1} \\ 1 \end{bmatrix}$ for $(\psi, g) \in \mathbb{R}^n \times \mathbb{R}^2$.

Step 5 Compute $\eta = (\eta_1, \eta_2)^*$, $M = (m_{ij})_{i,j=1}^2$, where

$$\eta_i = \psi^* f_{xx}[v_i, \widehat{x}] \tag{53}$$

$$m_{ij} = \psi^* f_{xx}\left[v_i, v_j\right]. \tag{54}$$

Step 6 Solve $M\xi = g - \eta$ for $\xi = (\xi_1, \xi_2)^*$

Step 7 Set $x_{\text{new}} = x + \widehat{x} + \xi_1 v_1 + \xi_2 v_2$

Step 8 Set $w_{\text{bor,new}} = \frac{\psi}{\|\psi\|}$

Step 9 Set $v_{1,\text{bor,new}} = \frac{v_1}{\|v_1\|}$,

$$v_{2,\text{bor,new}} = v_2 - (v_2^* v_{1,\text{bor,new}}) v_{1,\text{bor,new}}, \qquad v_{2,\text{bor,new}} = \frac{v_{2,\text{bor,new}}}{\left\|v_{2,\text{bor,new}}\right\|}.$$

Output: $x_{\text{new}} = (u_{\text{new}}, \alpha_{\text{new}})$

*Choosing initial parameters and approximations.* Choose $\mu = 0$ and $w_{\text{bor}}$ from $f_u^* w_{\text{bor}} = \lambda w_{\text{bor}}$, $\|w_{\text{bor}}\| = 1$, where $\lambda$ is the smallest real eigenvalue in absolute value of $f_u^*$. Given the tangent vector $v = v(x)$ to the solution curve, choose

$$v_{1,\text{bor}} = \frac{v}{\|v\|},$$
$$v_{2,\text{bor}} = \frac{w - (w^* v)v}{\|w - (w^* v)v\|},$$

where $f_x w = \lambda w$, and $\lambda$ is the real eigenvalue of $f_x$ with smallest absolute value. We also use the approximations:

$$f_{xx}[v, v] \approx \frac{\|v\|^2}{\delta^2}\left[f\left(x + \delta\frac{v}{\|v\|}\right) - 2f(x) + f\left(x - \delta\frac{v}{\|v\|}\right)\right], \tag{55}$$

$$f_{xx}[v, w] \approx \frac{\|v\|\,\|w\|}{\delta^2}\left[f\left(x + \delta\frac{v}{\|v\|} + \delta\frac{w}{\|w\|}\right) - f\left(x + \delta\frac{w}{\|w\|}\right) - f\left(x + \delta\frac{v}{\|v\|}\right) + f(x)\right]. \tag{56}$$

#### 4.3.2. Switching branches at simple branch points

When the discriminant of the *algebraic branching equation* (ABE) is greater than zero (see e.g. [26]), we have a *simple branch point* (BP); that is, two distinct solution branches of (2) pass through $x_0$.

Although branch points are not generic along equilibrium curves, they do appear in many applications due to symmetries. For instance, they appear in the Brusselator Example 1, because this example has a reflectional symmetry; the BP points are symmetry-breaking bifurcations in this case.

Branch points are symmetry-breaking bifurcations. Although they are not generic along equilibrium curves, they appear in many applications. For instance, they appear in the Brusselator Example 1, because this example has a reflectional symmetry.

Once a simple branch point $x_0 = x(s_0)$ on a solution branch $x(s)$ has been located, one may want to compute the bifurcating branch. Three methods of switching branches are implemented in CL_MATCONTL [18,19].

1. A branch switching method based on the ABE (see e.g. [26] and references there), which requires the computation of second order derivatives using (55), (56).
2. Computing the first solution point $x_1$ on the bifurcating branch from

$$f(x_1) = 0, \qquad (x_1 - x_0)^* v_2^0 - \Delta s = 0, \tag{57}$$

where $v_2^0$ is the second null vector of $f_x^0$, with $v_2^0 \perp v_1^0 = \frac{dx(s_0)}{ds}$, $\|v_1^0\| = \|v_2^0\| = 1$. This method is implemented in AUTO [31] and works well in many applications, although there may be situations where it fails.
3. An improved version of (57) using an 'angular' bisection-like procedure in $\mathcal{N}\left(f_x^0\right)$ spanned by $v_1^0$ and $v_2^0 \perp v_1^0$, where the initial angle is the one between $v_1^0$ and $v_2^0$; see [22].

## 5. Continuation of fold and Hopf bifurcations

### 5.1. Fold continuation

We again use the system (31) of $n+1$ scalar equations, but now for $n+2$ components $x = (u, \alpha) \in \mathbb{R}^n \times \mathbb{R}^2$. Again $g$ is obtained by solving (32), where $g_x$ is computed using (34), (35), and (37).

There are three generic codimension-2 bifurcations on a fold curve: *Bogdanov–Takens* (or *double zero*) points (BT), *zero-Hopf* points (ZH), and *cusp* points (CP); see e.g. [1]. These are detected and located by the corresponding modifications of CL_MATCONT test functions.

1. (BT) An additional real eigenvalue of $f_u$ meets the imaginary axis: $\lambda_{1,2} = 0$, with geometric multiplicity remaining one. The CL_MATCONT test function to detect Bogdanov–Takens points is $\psi_{BT}^M = \widehat{w}^* \widehat{v}$, where $f_u \widehat{v} = f_u^* \widehat{w} = 0$, $\widehat{w}^* \widehat{w} = \widehat{v}^* \widehat{v} = 1$. This is replaced in CL_MATCONTL by

$$\psi_{BT} := w^* v, \quad \text{where } Cv = C^* w = 0, \qquad w^* w = v^* v = 1. \tag{58}$$

2. (ZH) Two extra nonreal eigenvalues $\lambda_{2,3}$ meet the imaginary axis: $\lambda_1 = 0$, $\lambda_{2,3} = \pm i\omega_0$, $\omega_0 > 0$. To detect zero-Hopf points, we use two test functions obtained by a slight modification of the test functions (26) and (27) used to detect Hopf points. Let

$$\lambda_{\min}(s) := \min\{|(\lambda_i(s))| : 1 \le i \le m\} \tag{59}$$

be a numerical approximation of the zero eigenvalue on $f_u$ on a fold curve. We further assume that the set $\Lambda_1(s) = \{\lambda_i(s)\}_{i=1}^m$ is ordered in such a way that the eigenvalue $\lambda_k(s)$ with $|(\lambda_k(s))| = \lambda_{\min}(s)$ has the index $k = m$. We next define

$$M_u(s) := \text{card}\{\lambda_i(s) + \lambda_j(s) : 1 \le i < j \le m-1, \text{Re}(\lambda_i(s) + \lambda_j(s)) > 0\} \tag{60}$$

and a piecewise smooth function:

$$\mu_{\min}(s) := \min\left\{|(\lambda_i(s) + \lambda_j(s))| : 1 \le i < j \le m-1\right\}. \tag{61}$$

In our case:

$$\psi_{ZH}^{(1)}(x(s)) := \mu_{\min}(s)(-1)^{M_u(s)}, \tag{62}$$

$$\psi_{ZH}^{(2)}(x(s)) := (-1)^{m_u^{\text{pairs}}}. \tag{63}$$

Then a zero-Hopf point is detected as:

$$\text{ZH}: \ \psi_{ZH}^{(1)}(x(s_k))\,\psi_{ZH}^{(1)}(x(s_{k+1})) < 0 \quad \text{and} \quad \psi_{ZH}^{(2)}(x(s_k))\,\psi_{ZH}^{(2)}(x(s_{k+1})) < 0. \tag{64}$$

Note [9], $\psi_{ZH}^{(1)}$ is regular at a generic zero-Hopf point. However, $\psi_{ZH}^{(1)}$ will also vanish at Bogdanov–Takens points.
3. (CP) $f_u$ has one zero eigenvalue and no other eigenvalues on the imaginary axis, and the quadratic normal form coefficient (38) is $a = 0$. The test function for a cusp point coincides with (39):

$$\psi_{CP}(x(s)) := a_h. \tag{65}$$

### 5.2. Continuation of Hopf bifurcations

We use again the system (40) of $n+2$ scalar equations, but for $n+3$ components $(x, \kappa) = (u, \alpha, \kappa) \in \mathbb{R}^n \times \mathbb{R}^2 \times \mathbb{R}$, in this case. The size of the border is increased by one, see Remark 3.

There are four generic codimension-2 bifurcations on the curve of Hopf bifurcations: *Bogdanov–Takens* (or *double zero*) points (BT), *zero-Hopf* points (ZH), *double-Hopf* points (HH), and *generalized Hopf* points (GH); see e.g. [1]. These are detected and located by the corresponding modifications of CL_MATCONT test functions.

1. (BT) As in CL_MATCONT [9], we use
$$\psi_{BT} := \kappa \tag{66}$$
as the test function to detect Bogdanov–Takens points, where $\kappa$ is the last component of $(u, \alpha, \kappa) \in \mathbb{R}^n \times \mathbb{R}^2 \times \mathbb{R}$.
2. (ZH) To detect zero-Hopf points, we use the same test function (25) as we use to detect LP points:
$$\psi_{ZH}(x(s)) := \lambda_{\min}(s)(-1)^{m_u(s)}. \tag{67}$$
Note, by Lemma 3, $\psi_{ZH}$ is equivalent to the corresponding test function $\det(A(s))$ used in CL_MATCONT [9].
3. (HH) To detect double-Hopf points, we use two test functions obtained by a slight modification of the test functions (26) and (27) used to detect Hopf points. This modification is to ensure that the two eigenvalues, say, $\lambda_{k,k+1} = \pm i\omega_0,\ \omega_0 > 0$ (in exact arithmetic) do not contribute to our test functions. This is important from the numerical point of view, since $\Re(\lambda_k(s)) = \Re(\lambda_{k+1}(s))$ can numerically be nonzero, but small in absolute value, and may change sign during a continuation of Hopf bifurcations. Let
$$\Lambda_1'(s) := \{\lambda_i(s) \in \Lambda_1(s), i \neq k, k+1\}_{i=1}^{m}. \tag{68}$$
Define:
$$M_u'(s) := \operatorname{card}\{\lambda_i(s) + \lambda_j(s) : i < j, \lambda_i(s), \lambda_j(s) \in \Lambda_1'(s), \operatorname{Re}\lambda_i(s) + \operatorname{Re}\lambda_j(s) > 0\}, \tag{69}$$
$$\mu_{\min}'(s) := \min\left\{|\operatorname{Re}\lambda_i(s) + \operatorname{Re}\lambda_j(s)| : i < j, \lambda_i(s), \lambda_j(s) \in \Lambda_1'(s)\right\}. \tag{70}$$
Then we use
$$\psi_{HH}^{(1)}(x(s)) := \mu_{\min}'(s)(-1)^{M_u'(s)} \tag{71}$$
as the test function to detect double-Hopf bifurcations. By an argument analogous to that in Lemma 3, it is easy to see that $\psi_{HH}^{(1)}$ is equivalent to the test function
$$\begin{aligned}\psi_{HH}^{M}(x(s)) &:= \det\left[2A(s)\Big|_{\left(N\left(A^2+\kappa I_n\right)^T\right)^{\perp}} \odot I_{n-2}\right] \\ &= \prod_{1\le i<j\le n, i,j\neq k,k+1} \left(\operatorname{Re}\lambda_i(s) + \operatorname{Re}\lambda_j(s)\right)\end{aligned} \tag{72}$$
used in CL_MATCONT [9]. Note that
$$\psi_{HH}^{(2)}(x(s)) := (-1)^{m_u^{\text{pairs}}}, \tag{73}$$
changes sign, where $m_u^{\text{pairs}}(s)$ is the number of complex eigenvalues from $\Lambda_1'(s)$ with $\operatorname{Re}(\lambda_i(s)) \geq 0$ and $\operatorname{Im}(\lambda_i(s)) > 0$, when the number of unstable complex conjugate eigenpairs from $\Lambda_1'(s)$ changes.
4. (GH) We use an approximation $l_{1h}$ to the first Lyapunov coefficient $l_1$ (see (49), (50))
$$\psi_{GH}(x(s)) := l_{1h} \tag{74}$$
as the test function to detect a generalized Hopf bifurcation.

## 6. Examples

All computations below are performed on a laptop running 32-bit Windows XP and under 32-bit MATLAB version 2010a. The machine has a 2.67 GHz Intel Core i7 CPU with 4 GB of RAM. The computations are performed on a single core of the i7 processor. Timing is measured using the MATLAB profiler function which returns the actual CPU time. In all the tables below '$t$' stands for time in seconds.

The main goal of these examples is not to discover some new interesting bifurcation behaviors, but to verify that the new CIS-based algorithms can accurately, reliably, and within a reasonable time detect, locate, and continue singularities of interest in large systems. We will use very fine grids to verify that the algorithms presented remain accurate and do not break down for large numbers of unknowns, as well as for timing purposes; it is not implied that these are necessary for the bifurcation study. In fact, in all the examples below fairly coarse grids are sufficient for that purpose, which is the reason these examples are selected in the first place. Note, the size $m$ of the invariant subspace in the examples is between 5 and 12.

No comparison is given between the computational results with new CIS-based test functions/locators and those in CL_MATCONT, as CL_MATCONT was not really intended to deal with large equilibrium systems like discretized PDEs. For example, for a discretized PDE the Jacobian will probably be sparse, and a good sparse LU decomposition could be added to CL_MATCONT, but the determinant would be badly scaled. So this approach cannot be recommended.

**Table 1**
1D Brusselator. Continuation of an equilibrium branch and locating a Hopf point. The CPU times for continuation, the CIS algorithm, and H point location are displayed, along with the value of the parameter at the H point, for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $b$ at H |
|---|---|---|---|---|---|
| 400 | 20 | 1.911 | 0.938 | 0.251 | 17.2056 |
| 800 | 20 | 2.208 | 1.149 | 0.220 | 17.2056 |
| 3 200 | 20 | 5.972 | 3.303 | 0.546 | 17.2056 |
| 12 800 | 30 | 49.293 | 19.860 | 6.734 | 17.2056 |
| 25 600 | 30 | 71.849 | 41.066 | 6.953 | 17.2056 |

**Table 2**
1D Brusselator. Continuation of a Hopf branch and locating a zero-Hopf point. The CPU times for H branch continuation, the CIS algorithm, and ZH point location are displayed, along with the values of the parameters at the ZH point, for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $(b, d_1)$ at ZH |
|---|---|---|---|---|---|
| 400 | 35 | 17.468 | 10.890 | 2.156 | (17.207, 1.0133) |
| 800 | 35 | 23.389 | 14.721 | 1.531 | (1.7206, 1.0126) |
| 3 200 | 35 | 71.769 | 42.104 | 6.061 | (1.7206, 1.0125) |
| 12 800 | 35 | 272.203 | 151.408 | 14.189 | (1.7206, 1.0125) |
| 25 600 | 35 | 694.144 | 419.630 | 21.875 | (1.7206, 1.0124) |

**Example 1.** The 1D *Brusselator*

$$\begin{aligned} &\frac{d_1}{l^2}u'' - (b+1)u + u^2 v + a = 0, \qquad \frac{d_2}{l^2}v'' + bu - u^2 v = 0, \quad \text{in } \Omega = (0,1), \\ &u(0) = u(1) = a, \qquad v(0) = v(1) = \frac{b}{a} \end{aligned} \tag{75}$$

is a well-known model system [32] for autocatalytic chemical reactions with diffusion. The left hand side of the equilibrium equations in (75) has the form (2), which is the right hand side of the dynamical system (1). This problem exhibits a rich bifurcation scenario and has been used in the literature as a standard model for bifurcation analysis [33–38]. We utilize the second-order central difference discretization

$$f'' \approx \frac{1}{h^2}(f_{i-1} - 2f_i + f_{i+1}), \quad h = (N+1)^{-1},$$

with a uniform grid of $N$ points. Since there are two unknowns per grid point, the resulting discrete problem, which has dimension $n = 2N$, can be written in the form (2). This discretization of the Brusselator is used in a CL_MATCONT example [9]. We consider the system (75) with a constant initial equilibrium solution

$$\begin{cases} u(x) = a, \\ v(x) = \dfrac{b}{a}. \end{cases} \tag{76}$$

The initial values of the parameters are: $a = 4$, $b = 17.1$, $d_1 = 1.0$, $d_2 = 2.0$, and $l = 12$.

(a) The initial equilibrium solution (76) is continued in parameter $b$, and a Hopf (H) point is located. Table 1 displays the CPU times for continuation, the CIS algorithm, H point location, and the value of $b$ at the H point for different values of $n$.
(b) The Hopf branch (2-parameter locus of Hopf points) is continued in $(b, d_1)$ starting at $(b, d_1) = (17.2056, 1.0)$. A zero-Hopf (ZH) point is located. Table 2 displays the CPU times for continuation, the CIS algorithm, ZH point location, and the value of $(b, d_1)$ at the ZH point for different values of $n$. See Fig. 1(A) for the corresponding bifurcation diagram (for $n = 25\,600$).

**Remark 8.** Though the eigenvalues at the ZH point are (for $n = 1000$): $1.547271 \times 10^{-9}$, $7.188694 \times 10^{-15} - 3.859847\text{i}$, $7.188694 \times 10^{-15} + 3.859847\text{i}$, as they should be, a more careful analysis uncovers that this is actually a branch point (BP) on the curve of Hopf bifurcations. Detection of BPs on the curve of Hopf bifurcations is currently not included in CL_MATCONTL.

**Example 2.** *1D Brusselator* (75) with an approximate nonconstant initial equilibrium solution

$$\begin{cases} u(x) = a + 2\sin(\pi x), \\ v(x) = \dfrac{b}{a} - \dfrac{1}{2}\sin(\pi x). \end{cases} \tag{77}$$

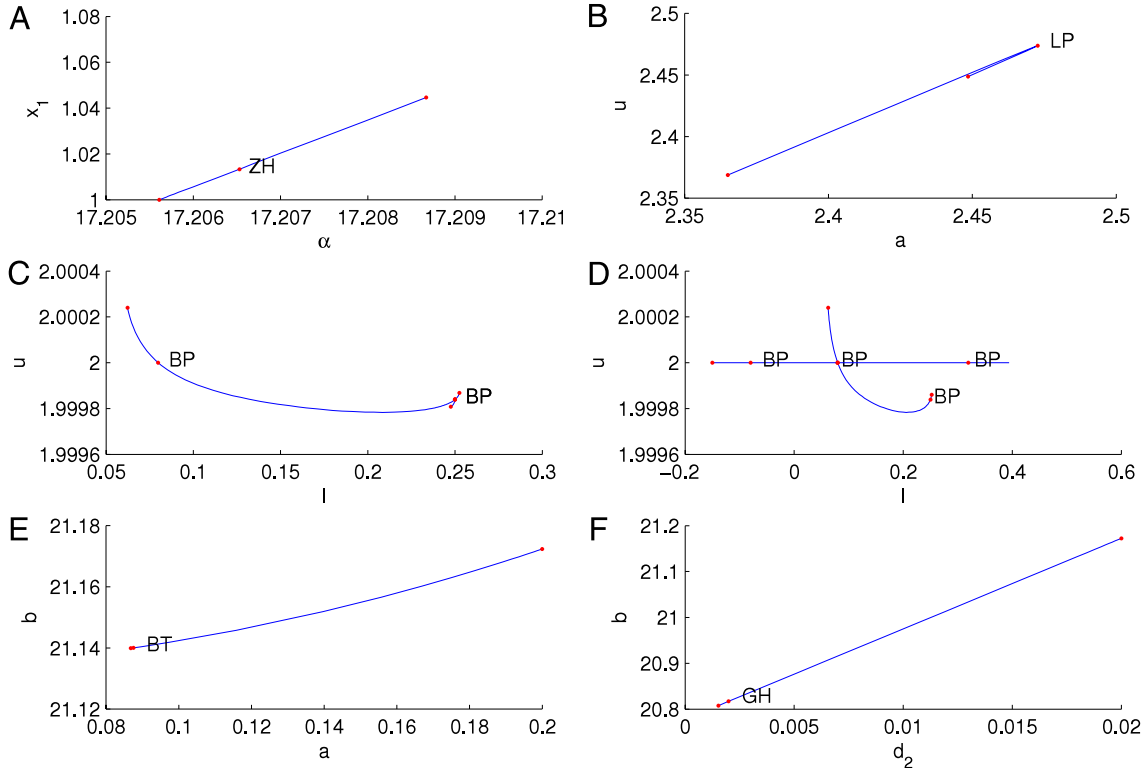


**Fig. 1.** A. Bifurcation diagram for a 1D Brusselator: continuation of a Hopf branch and locating a zero-Hopf point. B. Bifurcation diagram for a 1D Brusselator: continuation of an equilibrium branch and locating a fold point. C. Bifurcation diagram for a 1D Brusselator: switching from one branch of equilibria to another one. D. Bifurcation diagram for a 1D Brusselator: switching from one branch of equilibria to the second one and locating two branch points on the second branch. E. Bifurcation diagram for a 2D Brusselator: continuation of a Hopf branch and locating a Bogdanov–Takens point. F. Bifurcation diagram for a 2D Brusselator: continuation of a Hopf branch and locating a generalized Hopf point.

**Table 3**
1D Brusselator. Continuation of an equilibrium branch and locating a fold point. The CPU times for continuation, the CIS algorithm, and LP location are displayed, along with the value of the parameter at the LP, for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $a$ at LP |
|---|---|---|---|---|---|
| 800 | 20 | 2.283 | 1.006 | 0.529 | 2.4727 |
| 3 200 | 20 | 6.703 | 2.217 | 1.562 | 2.4727 |
| 12 800 | 20 | 26.18 | 8.06 | 7.124 | 2.4727 |
| 25 600 | 20 | 56.34 | 15.23 | 13.31 | 2.4727 |

**Table 4**
1D Brusselator. Continuation of an equilibrium branch and locating two branch points BP1 and BP2. The CPU times for continuation, the CIS algorithm, and BP1 and BP2 location are displayed, along with the values of the parameters at (BP1, BP2), for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $(l_1, l_2)$ at (BP1, BP2) |
|---|---|---|---|---|---|
| 800 | 55 | 6.216 | 2.203 | 0.703 | (0.079844, 0.24996) |
| 3 200 | 65 | 27.32 | 9.859 | 3.279 | (0.079844, 0.24996) |
| 12 800 | 65 | 138.9 | 35.88 | 50.48 | (0.079844, 0.24996) |
| 25 600 | 75 | 353.5 | 97.37 | 123.7 | (0.079845, 0.24988) |

The pair of functions (77) is not an equilibrium, but the continuer first locates an equilibrium close to this initial guess. The initial values of the parameters are: $a = 2.3$, $b = 4.6$, $d_1 = 0.0016$, $d_2 = 0.008$.

(a) The initial value of the parameter $l$ is 0.095. The initial equilibrium solution (77) is continued in parameter $a$, and a fold point (LP) is located at $a = 2.472741$. Table 3 displays the CPU times for continuation, the CIS algorithm, LP location, and the value of $a$ at the LP for different values of $n$. See Fig. 1(B) for the corresponding bifurcation diagram (for $n = 25\,600$).
(b) The initial value of the parameter $l$ is 0.06228. The initial equilibrium solution (77) is continued in parameter $l$, and two BPs (BP1 and BP2) are located. Table 4 displays the CPU times for continuation, the CIS algorithm, the BP1 and BP2 locations, and $(l_1, l_2)$ at (BP1, BP2) for different values of $n$. We next switch to another branch of equilibria at BP1 and BP2 and then continue both. Table 5 displays the CPU times for continuation, the CIS algorithm, and for switching from one branch of equilibria to another branch of equilibria at BP2 (using the bisection-like algorithm [22]) for different values

**Table 5**
1D Brusselator. Switching from one branch of equilibria to another branch of equilibria. The CPU times for continuation, the CIS algorithm, and switching from one branch of equilibria to another branch of equilibria are displayed for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | BP switchBisect $t$ (s) | Terminal $l$ |
|---|---|---|---|---|---|
| 800 | 10 | 1.668 | 0.393 | 0.079 | 2.475949 |
| 3 200 | 10 | 5.657 | 1.390 | 0.297 | 0.2475942 |
| 12 800 | 10 | 34.799 | 6.562 | 0.797 | 0.2468171 |
| 25 600 | 10 | 74.155 | 11.593 | 1.750 | 0.2475902 |

**Table 6**
2D Brusselator. Continuation of an equilibrium branch and locating a Hopf point. The CPU times for continuation, the CIS algorithm, and H point location are displayed, along with the parameter value at the H point, for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $b$ at H |
|---|---|---|---|---|---|
| 200 | 10 | 1.376 | 0.533 | 0.391 | 21.038 |
| 5 000 | 10 | 16.314 | 5.344 | 4.735 | 21.168 |
| 20 000 | 10 | 94.035 | 23.096 | 23.157 | 21.172 |

**Table 7**
2D Brusselator. Continuation of a Hopf branch and locating a Bogdanov–Takens point. The CPU times for H continuation, the CIS algorithm, and BT point location are displayed, along with the values of the parameters at the BT, for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $(a, b)$ at BT |
|---|---|---|---|---|---|
| 200 | 42 | 13.19 | 5.624 | 0.141 | (0.087208, 21.005) |
| 5 000 | 42 | 156.4 | 65.41 | 1.297 | (0.087510, 21.135) |
| 20 000 | 42 | 1539 | 670.1 | 7.907 | (0.087521, 21.140) |

of $n$. See Fig. 1C for the corresponding bifurcation diagram (for $n = 25\,600$) and Fig. 1D for the corresponding bifurcation diagram (for $n = 25\,600$) when switching from one branch of equilibria to another branch of equilibria at BP2 (using the algebraic branching equation, ABE).

**Example 3.** The *2D Brusselator* is

$$\begin{aligned} &\frac{d_1}{l^2}\Delta u - (b+1)u + u^2 v + a = 0, \qquad \frac{d_2}{l^2}\Delta v + bu - u^2 v = 0, \quad \text{in } \Omega = (0,1)\times(0,1), \\ &u|_{\partial\Omega} = a, \qquad v|_{\partial\Omega} = \frac{b}{a}. \end{aligned} \tag{78}$$

We utilize the second-order central difference discretization with uniform grid of $N$ grid points in each of the two directions. The resulting discrete problem, which has dimension $n = 2N^2$, can be written in the form (2). We consider the system (78) with a constant initial equilibrium solution

$$\begin{cases} u(x) = a, \\ v(x) = \dfrac{b}{a}. \end{cases} \tag{79}$$

The initial values of the parameters are: $a = 0.2$, $b = 22.1$, $d_1 = 1$, $d_2 = 0.02$, and $l = 1$.

(a) The initial equilibrium solution (79) is continued in parameter $b$, and a Hopf point is located. Table 6 displays the CPU times for continuation, the CIS algorithm, H point location, as well as the value of $b$ at the H point, for different values of $n$.
(b) The branch of Hopf bifurcations is continued in $(a, b)$ starting at $(a, b) = (0.2, 21.172)$, and a Bogdanov–Takens (BT) point is located at step 41. Table 7 displays the CPU times for continuation, the CIS algorithm, BT point location, and the value of $(a, b)$ at the BT point for different values of $n$. See Fig. 1(E) for the corresponding bifurcation diagram (for $n = 20\,000$).

**Remark 9.** The most accurate eigenvalues at the BT point we could get are $\pm 7.6 \times 10^{-6}$ (for $n = 200$) and $\pm 2.0 \times 10^{-5}$ (for $n = 20\,000$) with the residuals $4.8 \times 10^{-15}$ and $1.1 \times 10^{-12}$, respectively. Generically, there should also be a fold curve going through the BT point that is tangential to the curve of Hopf bifurcations in the projection on the parameter plane. However, no fold curve was found in this case.

**Table 8**
2D Brusselator. Continuation of a Hopf branch and locating a generalized Hopf point. The CPU times for H continuation, the CIS algorithm, and GH point location are displayed, along with the values of the parameters at the GH, for different values of $n$.

| $n$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $(d_2, b)$ at GH |
|---|---|---|---|---|---|
| 200 | 44 | 17.05 | 7.735 | 0.812 | (0.0020042, 20.685) |
| 5 000 | 44 | 168.0 | 62.97 | 9.771 | (0.0019933, 20.812) |
| 20 000 | 44 | 1205 | 414.5 | 143.2 | (0.0019929, 20.817) |

**Table 9**
2D Arch. Continuation of an equilibrium branch and locating a branch point BP. The CPU times for continuation, the CIS algorithm, and BP location, and the parameter values at the BP for different values of $(m, n, N)$ are displayed.

| $(m, n, N)$ | Steps | Total $t$ (s) | CIS $t$ (s) | Locator $t$ (s) | $\alpha$ at BP |
|---|---|---|---|---|---|
| (3, 48, 1353) | 20 | 26.36 | 1.720 | 9.277 | 2.09987 |
| (6, 96, 5013) | 20 | 271.4 | 24.87 | 142.8 | 2.10363 |
| (12, 192, 19 245) | 20 | 1768 | 233.2 | 947.3 | 2.10681 |

(c) The branch of Hopf bifurcations is continued in two parameters $(d_2, b)$ starting at $(d_2, b) = (0.02, 21.172)$, and a generalized Hopf (GH) point is located at step 42. The value of the first Lyapunov coefficient (74) is $\psi_{GH} = -1.7 \times 10^{-6}$ (for $n = 20\,000$) at the GH point, and it changes sign from positive to negative at the GH point. Table 8 displays the CPU times for continuation, the CIS algorithm, GH point location, and the value of $(d_2, b)$ at the GH point for different values of $n$. See Fig. 1(F) for the corresponding bifurcation diagram (for $n = 20\,000$).

**Example 4.** *Deformation of a 2D arch.* We consider the snap-through of an elastic arch, shown in Fig. 2. The arch is pinned at both ends, and the $y$ displacement of the center of the arch is controlled as a continuation parameter.

Let $\Omega_0 \subset \mathbb{R}^2$ be the interior of the undeformed arch (Fig. 2, top left), and write the boundary as $\Gamma = \Gamma_D \cup \Gamma_N$, where $\Gamma_D$ consists of the two points where the arch is pinned, and $\Gamma_N$ is the remainder of the boundary, which is free. At equilibrium, material points $X \in \Omega_0$ in the deformed arch move to positions $x = X + u$. Except at the control point $X_{\text{center}}$ in the center of the arch, this deformation satisfies the equilibrium force-balance equation [39]

$$\sum_{J=1}^{2} \frac{\partial S_{IJ}}{\partial X_J} = 0, \quad X \in \Omega_0,\ I = 1, 2 \tag{80}$$

where the second Piola–Kirchhoff stress tensor $S$ is a nonlinear function of the Green strain tensor $E$, where $E := \frac{1}{2}(F^T F - I)$, $F := \frac{\partial u}{\partial X}$. Eq. (80) is thus a fully nonlinear second-order elliptic system. The boundary and the control point $X_{\text{center}}$ are subject to the boundary conditions

$$u = 0 \quad \text{on } \Gamma_D, \qquad SN = 0 \quad \text{on } \Gamma_N, \text{ where } N \text{ is an outward unit normal,} \tag{81}$$

$$e_2 \cdot u = \alpha, \qquad e_1 \cdot (FSN) = 0. \tag{82}$$

The first condition at $X_{\text{center}}$ says that the vertical displacement is determined; the second condition says that there is zero force in the horizontal direction.

We discretize (80) with biquadratic isoparametric Lagrangian finite elements. Let $m$ be the number of elements through the arch thickness, and $n$ be the number of elements along the length of the arch; then there are $(2m + 1)(2n + 1)$ nodes, each with two associated degrees of freedom. The Dirichlet boundary conditions are used to eliminate four unknowns, and one of the unknowns (written as $a$) is used as a control parameter. Specifically, we choose the continuation parameter $a$ to be the $y$ displacement of the node in middle of the arch. So the resulting discrete problem, which has dimension $N = 2(2m + 1)(2n + 1) - 5$, can be written in the form (2). We consider the system (80)–(82) with a constant initial equilibrium solution

$$u = 0. \tag{83}$$

The initial value of the parameter $a$ is 0.

The initial equilibrium solution (83) is continued in parameter $a$, and a BP is located. Fig. 2 displays the results for $(m, n, N) = (4, 60, 2173)$ (top left, top right, and bottom left) and for $(m, n, N) = (12, 192, 19\,245)$ (bottom right).

Table 9 displays the CPU times for continuation, the CIS algorithm, and BP location, along with $a$ at the BP, for different values of $(m, n, N)$. We next switch to another branch of equilibria at the BP and then continue it. Table 10 displays the CPU times for continuation, the CIS algorithm, and switching from one branch of equilibria to another branch of equilibria for different values of $(m, n, N)$ See Fig. 2, bottom right for the corresponding bifurcation diagram (for $(m, n, N) = (12, 192, 19\,245)$). The results presented here are mostly new. A preliminary result, a BP location for $(m, n, N) = (4, 60, 2173)$, is presented in [20].

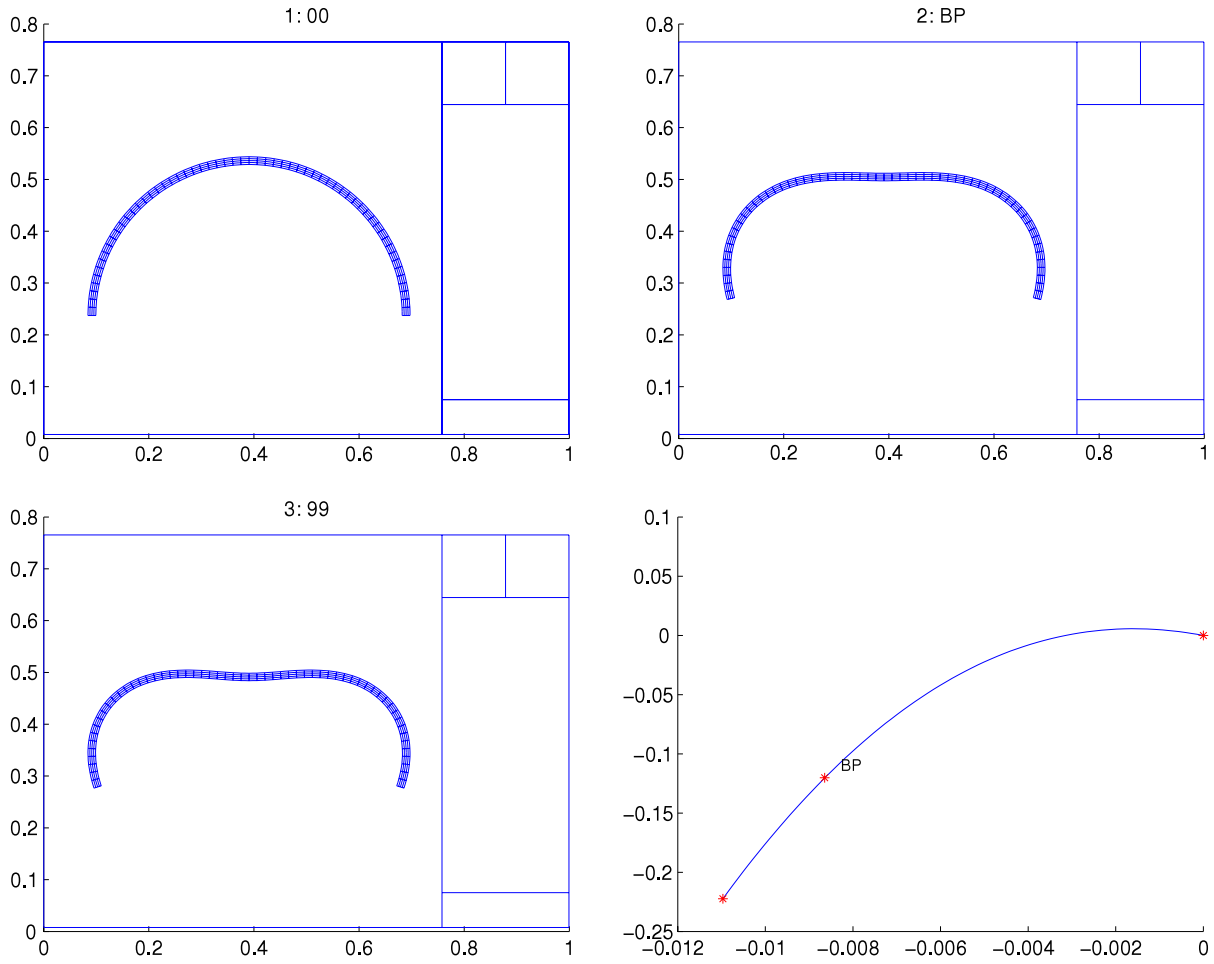


**Fig. 2.** Top left: the undeformed arch, top right: the arch at the bifurcation point, bottom left: the arch at the end of continuation along the original branch, bottom right: the bifurcation diagram.

**Table 10**
2D Arch. Switching from one branch of equilibria to another branch of equilibria. The CPU times for continuation, the CIS algorithm, and switching from one branch of equilibria to another branch of equilibria are displayed for different values of $(m, n, N)$.

| $(m, n, N)$ | Steps | Total $t$ (s) | CIS $t$ (s) | BPswitchBisect $t$ (s) | Terminal $\alpha$ |
|---|---|---|---|---|---|
| (3, 48, 1353) | 20 | 8.915 | 1.720 | 0.829 | 2.28117 |
| (6, 96, 5013) | 20 | 197.9 | 16.10 | 2.864 | 2.14750 |
| (12, 192, 19 245) | 20 | 522.9 | 104.2 | 20.20 | 2.11831 |

## Acknowledgments

We thank the two reviewers for their careful reading of our manuscript and for their many useful questions, remarks, and suggestions that have led to a substantial improvement of the paper.

The author was supported in part under NSF DMS-0209536 and NSF ATM-0417774.

## References

[1] Yu.A Kuznetsov, Elements of Applied Bifurcation Theory, third ed., Springer-Verlag, New York, 2004.
[2] G.M. Shroff, H.B. Keller, Stabilization of unstable procedures: the recursive projection method, SIAM J. Numer. Anal. 30 (1993) 1099–1120.
[3] C.S. Chien, M.H. Chen, Multiple bifurcations in a reaction–diffusion problem, Comput. Math. Appl. 35 (1998) 15–39.
[4] E.J. Doedel, H. Sharifi, Collocation methods for continuation problems in nonlinear elliptic PDEs, issue on continuation, in: Methods in Fluid Mechanics, in: Notes on Numer. Fluid. Mech., vol. 74, 2000, pp. 105–118.
[5] K.A. Cliffe, A. Spence, S.J. Tavener, The numerical analysis of bifurcations problems vith application to fluid mechanics, Acta Numer. (2000) 1–93.
[6] H.A. Dijkstra, F.W. Wubs, A.K. Cliffe, E. Doedel, I.F. Dragomirescu, B. Eckhardt, A.Y. Gelfgat, A.L. Hazel, V. Lucarini, A.G. Salinger, E.T. Phipps, J. Sanchez-Umbria, H. Schuttelaars, L.S. Tuckerman, U. Thiele, Numerical bifurcation methods and their application to fluid dynamics: analysis beyond simulation, Commun. Comput. Phys. 15 (2014) 1–45.
[7] A.G. Salinger, E.A. Burroughs, R.P. Pawlowski, E.T. Phipps, L.A. Romero, Bifurcation tracking algorithms and software for large scale applications, Int. J. Bifurcation Chaos 15 (2005) 1015–1032.

[8] H. Uecker, D. Wetzel, J.D. Rademacher, A MATLAB package for continuation and bifurcation in 2D elliptic systems, in: Numerical Mathematics: Theory, Methods and Applications, 2013, in print. http://www.staff.uni-oldenburg.de/hannes.uecker/pde2path/p2p.pdf.
[9] A. Dhooge, W. Govaerts, Yu.A Kuznetsov, W. Mestrom, A.M. Riet, MATLAB continuation software package CL_MATCONT, December. 2008. http://sourceforge.net/projects/matcont/.
[10] A. Dhooge, W. Govaerts, Yu.A Kuznetsov, MATCONT: a MATLAB package for numerical bifurcation analysis of odes, ACM Trans. Math. Softw. 29 (2003) 141–164.
[11] W. Govaerts, Numerical Methods for Bifurcations of Dynamical Equilibria, SIAM, Philadelphia, 2000.
[12] J.W. Demmel, L. Dieci, M.J. Friedman, Computing connecting orbits via an improved algorithm for continuing invariant subspaces, SIAM J. Sci. Comput. 22 (2001) 81–94.
[13] L. Dieci, M.J. Friedman, Continuation of invariant subspaces, Numer. Linear Algebra Appl. 8 (2001) 317–327.
[14] M.J. Friedman, Improved detection of bifurcations in large nonlinear systems via the Continuation of Invariant Subspaces algorithm, Internat. J. Bifur. Chaos 11 (2001) 2277–2285.
[15] D. Bindel, J. Demmel, M. Friedman, Continuation of invariant subspaces for large bifurcation problems, in: Proceedings of the SIAM Conference on Linear Algebra, Williamsburg, VA, 2003.
[16] D. Bindel, J. Demmel, M. Friedman, Continuation of invariant subspaces for large bifurcation problems, SIAM J. Sci. Comput. 30 (2008) 637–656.
[17] W.-J. Beyn, W. Kleß, V. Thümmler, Continuation of low-dimensional invariant subspaces in dynamical systems of large dimension, in: B. Fiedler (Ed.), Ergodic Theory, Analysis, and Efficient Simulation of Dynamical Systems, Springer, 2001, pp. 47–72.
[18] D. Bindel, M. Friedman, W. Govaerts, J. Hughes, Yu.A Kuznetsov, CL_MATCONTL continuation toolbox in MATLAB, January. 2010. http://webpages.uah.edu/~hughesjs/.
[19] D. Bindel, M. Friedman, W. Govaerts, J. Hughes, Yu.A Kuznetsov, CL_MATCONTL tutorial, Januray 2010. http://webpages.uah.edu/~hughesjs/.
[20] D. Bindel, W. Demmel, M. Friedman, W. Govaerts, Yu.A 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.
[21] M. Friedman, W. Govaerts, Yu.A Kuznetsov, B. Sautois, Continuation of homoclinic orbits in MATLAB, in: Proceedings of the ICCS conference 2005, Volume 3514/2005, Atlanta, GA, 2005, pp. 263–270.
[22] J. Hughes, M. Friedman, A bisection-like algorithm for branch switching at a simple branch point, J. Sci. Comput. 41 (2009) 62–69.
[23] V.A. Selivanov, M. Cascante, M. Friedman, M.F. Schumaker, M. Trucco, T.V. Votyakova, Multistationary and oscillatory modes of free radicals generation by the mitochondrial respiratory chain revealed by a bifurcation analysis, PLoS Comput. Biol. (2012) e1002700.
[24] W. Govaerts, Stable solvers and block elimination for bordered systems, SIAM J. Matrix Anal. Appl. 12 (1991) 469–483.
[25] W. Govaerts, J.D. Pryce, Mixed block elimination for linear systems with wider borders, IMA J. Numer. Anal. 13 (1993) 161–180.
[26] W.-J. Beyn, A. Champneys, E.J. Doedel, Yu.A Kuznetsov, B. Sandstede, W. Govaerts, Numerical continuation and computation of normal forms, in: B. Fiedler (Ed.), Handbook of Dynamical Systems III: Towards Applications, Elsevier, 2001 (Chapter 4).
[27] M. Friedman, W. Qiu, On the location and continuation of Hopf bifurcations in large-scale problems, Internat. J. Bifur. Chaos 18 (2008) 1589–1597.
[28] A. Griewank, G. Reddien, Characterization and computation of generalized turning points, SIAM J. Numer. Anal. 21 (1984) 176–185.
[29] E. Allgower, H. Schwetlick, A general view of minimally extended systems for simple bifurcation points, ZAMM Z. Angew. Math. Mech. 77 (1997) 83–98.
[30] E. Allgower, K. Georg, Numerical path following, in: P.G. Ciarlet, J.L. Lions (Eds.), Handbook of Numerical Analysis, Volume 5, North-Holland, 1997, pp. 3–207.
[31] E. Doedel, A. Champneys, T. Fairgrieve, Yu.A Kuznetsov, B. Sandstede, X.-J. Wang, AUTO97: continuation and bifurcation software for ordinary differential equations (with homcont), 1997. http://indy.cs.concordia.ca/.
[32] R. Lefever, I. Prigogine, Symmetry-breaking instabilities in dissipative systems II, J. Chem. Phys. 48 (1968) 1695–1700.
[33] D. Schaeffer, M. Golubitsky, Bifurcation analysis near a double eigenvalue of a model chemical reaction, Arch. Ration. Mech. Anal. 75 (1981) 315–347.
[34] M. Golubitsky, D.G. Schaeffer, Singularities and Groups in Bifurcation Theory, Vol. 1, in: Applied Mathematical Sciences, vol. 51, Springer-Verlag, New York, 1985.
[35] G. Dangelmayr, Degenerate bifurcations near a double eigenvalue in the Brusselator, J. Aust. Math. Soc. Ser. B 28 (1987) 486–535.
[36] P. Ashwin, Z. Mei, A Hopf bifurcation with Robin boundary conditions, J. Dynam. Differential Equations 6 (1994) 487–503.
[37] C.S. Chien, Z. Mei, C.L. Shen, Numerical continuation at double bifurcation points of a reaction–diffusion problem, Internat. J. Bifur. Chaos 8 (1997) 117–139.
[38] Z. Mei, Numerical bifurcation analysis for reaction–diffusion equations, Ph.D. Thesis, University of Marburg, 1997.
[39] O. Zienkiewicz, R.T. Taylor, The Finite Element Method: Volume 2, Solid Mechanics, fifth ed., Butterworth Heinemann, Oxford, 2000.