Numerical Continuation of Connecting Orbits of Maps in MATLAB

R. Khoshsiar Ghaziani¹,², W. Govaerts², Yu.A. Kuznetsov³, and H.G.E. Meijer⁴

¹ Department of Mathematics, Shahrekord University, P.O. Box 115, Shahrekord, Iran
² Department of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281-S9, B-9000 Gent, Belgium
³ Department of Mathematics, Utrecht University, P.O. Box 80010, 3508 TA Utrecht, The Netherlands
⁴ Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

(November 2007)

We present new or improved methods to continue heteroclinic and homoclinic orbits to fixed points in iterated maps and to compute their fold bifurcation curves, corresponding to the tangency of the invariant manifolds. The proposed methods are applicable to general n-dimensional maps and are implemented in MATLAB. They are based on the continuation of invariant subspaces (CIS) algorithm, which is presented in a novel way. The systems of defining equations include the Riccati equations appearing in CIS for bases of the generalized stable and unstable eigenspaces. We use the bordering techniques to continue the folds, and provide full algorithmic details on how to treat the Jacobian matrix of the defining system as a sparse matrix in MATLAB.

For a special – but important in applications – case n = 2 we describe the first MATLAB implementation of known algorithms to grow one-dimensional stable and unstable manifolds of the fixed points of noninvertible maps.

The methods are applied to study heteroclinic and homoclinic connections in the generalized Hénon map.

1 Introduction

The accurate computation of orbits connecting fixed points of an iterated map, and the study of associated topological properties have long been recognized as a very important problem both in the theory of nonlinear dynamical systems and in a variety of applied problems, e.g. in models for economical, biological, and physical phenomena. Indeed, as discovered by Poincaré and Birkhoff, such orbits may generate rich dynamics. For example, an orbit that connects a
hyperbolic fixed point to itself (a homoclinic orbit) generically implies the existence of an infinite number of periodic orbits nearby, see [24,30,31] and tutorial presentations in [17,26,32]. As discovered in [9,10,16], the appearance of a pair of such homoclinic orbits is accompanied by an infinite sequence of fold and period-doubling bifurcations of periodic orbits, for more details see [25,27], as well as [22]. Moreover, since a homoclinic orbit of a planar map belongs to the intersection of the stable and the unstable invariant curves of a saddle fixed point, such orbits can be involved in the destruction of a closed invariant curve which is born, for example, at a Neimark-Sacker bifurcation [23,28,29]. This destruction mechanism has been studied in [1,4].

Numerical methods for bifurcation analysis of maps have received considerable attention recently and are supported by existing software. Location, analysis, and continuation of fixed-point and cycle bifurcations are implemented, e.g. in CONTENT [13] and Cl_MatContM [15]. Algorithms for the computation of the one-dimensional manifolds are implemented in DSTOOL [8,20] and DYNAMICS [34,35], while those for the continuation of homoclinic orbits and their tangencies using the projection asymptotic boundary conditions [2] are implemented in an AUTO-driver [33].

In this paper, we present new or improved methods to continue heteroclinic and homoclinic orbits to fixed points in iterated maps and to compute the tangencies of their invariant manifolds. In contrast with [33], the projection asymptotic boundary conditions are formulated using the continuation of invariant subspaces method [7], which we present in a novel way, using only linear algebra arguments. Moreover, in our approach the Riccati equation is included into the defining system and the continuation of the tangencies is based on the bordering technique. We revisit known algorithms to grow stable and unstable invariant 1D manifolds of cycles in a 2D map without using its inverse [8,20]. We discuss the implementation of all mentioned above methods in MATLAB as a part of Cl_MatContM. In particular, we provide full algorithmic details on how to treat the Jacobian matrix of the defining system as a sparse matrix in MATLAB. Therefore Cl_MatContM becomes a unique integrated toolbox for the numerical bifurcation analysis of maps and difference equations, that covers both local and global phenomena. To illustrate its strength, we compute heteroclinic and homoclinic connections in the generalized Hénon map.

We consider the $J$-th iterate of a smooth map at some parameter as follows:

$$x \mapsto f^J(x, \alpha) = f(f(\cdots f(x, \alpha), \alpha), \alpha), \quad f : \mathbb{R}^n \times \mathbb{R}^{n_\alpha} \to \mathbb{R}^n. \quad (1)$$
A sequence \((x_k)_{k \in \mathbb{Z}}\) is called a **connecting orbit** of the map \(f^J(\cdot, \alpha)\) at \(\alpha = \tilde{\alpha}\) if

\[
\begin{align*}
\lim_{k \to -\infty} x_k &= x_{-\infty}, \\
f^J(x_k, \tilde{\alpha}) &= x_{k+1}, \quad \text{for all } k \in \mathbb{Z}, \\
\lim_{k \to +\infty} x_k &= x_{+\infty}.
\end{align*}
\]

Due to the continuity, points \(x_{-\infty}\) and \(x_{+\infty}\) are **fixed points** of \(f^J(\cdot, \tilde{\alpha})\). A connecting orbit is called **homoclinic** if \(x_{-\infty} = x_{+\infty}\) and **heteroclinic** otherwise. From a geometrical point of view, the connecting orbit lies in the intersection of the unstable manifold \(W^u_{-\infty}\) of \(x_{-\infty}\) and the stable manifold \(W^s_{+\infty}\) of \(x_{+\infty}\). A connecting orbit is **regular** if \(x_{-\infty}\) and \(x_{+\infty}\) are hyperbolic and the stable manifold \(W^u_{-\infty}\) and the unstable manifold \(W^s_{+\infty}\) have transversal intersections at \(x_k\) for all \(k \in \mathbb{Z}\).

Degenerate cases occur when either one of its fixed points becomes nonhyperbolic or the connecting orbit loses transversality. In the former case the unstable and center-stable manifolds have a transversal intersection, which produces a connecting orbit with a singular endpoint. In the simplest case there is precisely one multiplier 1 or \(-1\), or one conjugate pair of multipliers of \(f^J(x, \alpha)\) on the unit circle. This gives us the **saddle-fold**, **saddle-flip**, **saddle-Neimark-Sacker** connecting orbits, respectively, see e.g. \([3, 18]\). We will not treat such cases and deal only with the latter case, i.e. the case of non-transversality.

The corresponding numerical problem, for a regular heteroclinic connection between hyperbolic fixed points \(x_1\) and \(x_N\) of \((1)\), is that of finding a solution \((x_k)_{k=1,2,\ldots,N}\) of the following system \([2]\):

\[
\begin{align*}
x_1 &= f^J(x_1, \alpha), \\
x_{k+1} &= f^J(x_k, \alpha), \quad k = 2, \ldots, N - 2, \\
x_N &= f^J(x_N, \alpha),
\end{align*}
\]

such that \((x_k)_{k=2,\ldots,N-1}\) leave \(x_1\) along its unstable manifold and approach \(x_N\) along its stable manifold. These requirements are then substituted by **projection boundary conditions** which place \(x_2\) and \(x_{N-1}\) into the corresponding tangent spaces, i.e. unstable and stable invariant subspaces of the Jacobian matrices of \(f^J\) at \(x_1\) and \(x_N\), respectively \([2]\).

## 2 Continuation of heteroclinic connections

We use an improved algorithm \([2]\) for locating and continuing connecting orbits, which is based on an algorithm for the continuation of invariant subspaces.
(CIS) [5,7]. Assume the eigenvalues of the Jacobian matrices \( f^J(x_1, \alpha) \) \_x \_1 \_1 and \( f^J(x_N, \alpha) \) \_x \_N \_N, where the subscript denotes differentiation, are ordered, respectively, as follows:

\[
|\lambda^U_1| \leq \ldots \leq |\lambda^U_{n_U}| < 1 < |\lambda^U_1| \leq \ldots \leq |\lambda^U_{n_U}|,
\]

\[
|\lambda^S_1| \leq \ldots \leq |\lambda^S_{n_S}| < 1 < |\lambda^S_{n_S}| \leq \ldots \leq |\lambda^S_{n_S}|.
\]

The algorithm requires the evaluation of various projections associated with the eigenspaces of \( f^J(x_1, \alpha) \) \_x \_1 \_1 \( f^J(x_N, \alpha) \) \_x \_N \_N. These projections are constructed using the real Schur factorizations.

\[
(f^J(x_1, \alpha))_x = Q^{(1)} R^{(1)} [Q^{(1)}]^T, \quad (f^J(x_N, \alpha))_x = Q^{(N)} R^{(N)} [Q^{(N)}]^T,
\]

where \( Q^{(1)}, R^{(1)}, Q^{(N)} \) and \( R^{(N)} \) are \( n \times n \)-matrices.

The first factorization has been chosen so that the first \( n_U \) columns \( q^U_1, \ldots, q^U_{n_U} \) of \( Q^{(1)} \) form an orthonormal basis of the right invariant subspace \( S_1 \) of \( f^J(x_1, \alpha) \) \_x \_1 \_1, corresponding to \( \lambda^U_1, \ldots, \lambda^U_{n_U} \) and the remaining \( n - n_U \) columns \( q^U_{n_U+1}, \ldots, q^U_{n} \) of \( Q^{(1)} \) form an orthonormal basis of the orthogonal complement \( S_1^\perp \). Similarly, the first \( n_S \) columns \( q^S_1, \ldots, q^S_{n_S} \) of \( Q^{(N)} \) form an orthonormal basis of the right invariant subspace \( S_N \) of \( f^J(x_N, \alpha) \) \_x \_N \_N, corresponding to \( \lambda^S_1, \ldots, \lambda^S_{n_S} \) and the remaining \( n - n_S \) columns \( q^S_{n_S+1}, \ldots, q^S_{n} \) of \( Q^{(N)} \) form an orthonormal basis of the orthogonal complement \( S_N^\perp \).

When dealing with heteroclinic connections, we want to find a sequence of points \( (x_k)_{k=1,...,N} \) satisfying:

- Stationary state conditions for the initial fixed point:
  \[
  f^J(x_1, \alpha) - x_1 = 0;
  \]  

- The iteration conditions
  \[
  f^J(x_k, \alpha) - x_{k+1} = 0, \quad k = 2, 3, \ldots, N - 2;
  \]

- Stationary state conditions for the final fixed point:
  \[
  f^J(x_N, \alpha) - x_N = 0;
  \]

- The left projection boundary conditions
  \[
  (x_2 - x_1)^T q^U_{n_U+i} = 0, \quad i = 1, \ldots, n - n_U;
  \]
The right projection boundary conditions

\[(x_{N-1} - x_N)^T q_{n_S+i}^S = 0, \quad i = 1, \ldots, n - n_S.\]  

(8)

A regular zero of a system of equations (4), (5), (6), (7), and (8) corresponds to an approximation of a transversal heteroclinic orbit with hyperbolic fixed points. Thus, a zero for this system can be continued in one parameter, for example, using the standard pseudo-arclength continuation algorithm [19].

In the computational process the conditions in (7) and (8) imply that we need to access the unstable and stable eigenspaces of the linearization of the map (1) at the fixed points \(x_1\) and \(x_N\) at each step of the continuation, respectively. It is not efficient to recompute these spaces from scratch in each continuation step. In the next section we explain an algorithm for continuing the invariant subspaces \(S_1\) and \(S_2\) effectively. Contrary to [5,7], our algorithm is purely based on linear algebra arguments.

2.1 Continuation of invariant subspaces

Let \(A(\alpha) \in \mathbb{R}^{n \times n}\) denote \((f^J(x_1, \alpha))_x\). The basic continuation algorithm requires at each pseudo-arclength continuation step the computation of the orthogonal complement of the right invariant (unstable) \(n_U\)-dimensional subspace \(S(\alpha)\) of \(A(\alpha)\). In general, the function \(A(\alpha)\) is smooth in \(\alpha\), and it is important that \(S(\alpha)\) is smooth as well, as otherwise convergence difficulties can be expected. We show how to constructively obtain smooth bases for the unstable eigenspace and its orthogonal complement.

Continuation of invariant subspaces was introduced in [5]. We formulate it in a novel way, using only linear algebra arguments. To justify our construction, we recall that in our continuation procedure we parametrize a solution branch in terms of so called pseudo-arclength; let \(s\) denote the pseudo-arclength variable. Thus, both fixed points \(x_1\) and \(x_N\) as well as the parameter(s) \(\alpha\) are smooth functions of \(s\). The matrix-valued function \(A(\alpha) : \alpha \in \mathbb{R}^m \rightarrow \mathbb{R}^{n \times n}\) can thus be viewed as a smooth function of \(s \in \mathbb{R}\). As a consequence, we consider the continuation of invariant subspaces with respect to the scalar pseudo-arclength variable \(s\). For this reason, we use the notation \(A(s)\) for \(A(\alpha)\).

We first consider \(x_1\) and its unstable eigenspace. Suppose that initially we have the (real) block Schur factorization

\[A(0) = Q(0)R(0)Q^T(0), \quad Q(0) = [Q_1(0) \quad Q_2(0)],\]  

(9)

where \(A(0), R(0)\) and \(Q(0)\) are \(n \times n\)-matrices, \(Q(0)\) is orthogonal, \(Q_1(0)\) has
dimensions \( n \times n \) and \( R(0) \) is block upper triangular

\[
R(0) = \begin{bmatrix}
R_{11}(0) & R_{12}(0) \\
0 & R_{22}(0)
\end{bmatrix},
\]

where \( R_{11}(0) \) and \( R_{22}(0) \) are \( n_U \times n_U \)- and \((n-n_U) \times (n-n_U)\)-matrices, respectively; \( R_{ii}(0), i = 1, 2 \), are not required to be triangular. The columns of \( Q_1(0) \) span the unstable invariant subspace \( S(0) \) of \( A(0) \), and the columns of \( Q_2(0) \) span the orthogonal complement \( S^\perp(0) \). We want to obtain a block Schur factorization for the matrix \( A(s) \), close to \( A(0) \).

Suppose that the matrix \( A(s) \) has two groups of eigenvalues, \( \Lambda_1(s) \) (with modulus \( > 1 \)) and \( \Lambda_2(s) \) (with modulus \( < 1 \)), which stay disjoint for all \( s \) around 0. Then, in a neighborhood of \( s = 0 \), we need a smooth factorization

\[
A(s) = Q(s)R(s)Q^T(s), \quad Q(s) = [Q_1(s) \quad Q_2(s)],
\]

where \( R(s) \) is in block Schur form

\[
R(s) = \begin{bmatrix}
R_{11}(s) & R_{12}(s) \\
0 & R_{22}(s)
\end{bmatrix}.
\]

Here, \( R_{11} \) has eigenvalues \( \Lambda_1(s) \) and \( R_{22} \) has eigenvalues \( \Lambda_2(s) \). As shown in [6], it is always possible to obtain a smooth path of block Schur factorizations that satisfies (11) and (12). However, this smooth path is usually not unique.

Thus we can write

\[
Q(s) = Q(0)U(s), \quad \text{with} \quad U(0) = I,
\]

so that we only need to compute the \( n \times n \)-matrix \( U(s) \). We partition \( U(s) \) in blocks of the same size as \( R(0) \) in (10):

\[
U(s) = [U_1(s) \quad U_2(s)] = \begin{bmatrix}
U_{11}(s) & U_{12}(s) \\
U_{21}(s) & U_{22}(s)
\end{bmatrix},
\]

so that \( U_{11}(s) \) and \( U_{22}(s) \) are \( n_U \times n_U \)- and \((n-n_U) \times (n-n_U)\)-matrices, respectively.

We now show that we can always assume that \( U_{11}(s) \) and \( U_{22}(s) \) are symmetric positive-definite by redefining \( Q(s) \) and \( R(s) \) if necessary and that this defines \( Q(s) \) and \( R(s) \) in a unique way.

Proposition: Suppose \( Q(0) \) and \( R(0) \) are chosen such that (9) and (10) hold. Then for all \( s \) sufficiently close to 0 there exist a unique orthogonal matrix \( Q(s) = [Q_1, Q_2] \) of size \( n \times n \) such that the columns of \( Q_1 \) span the
unstable invariant subspace of $A(s)$ and the columns of $Q_2$ span the orthogonal complement of the unstable eigenspace, and a unique block triangular matrix

$$R(s) = \begin{bmatrix} R_{11}(s) & R_{12}(s) \\ 0 & R_{22}(s) \end{bmatrix}$$

(15)
of size $n \times n$ where $R_{11}$ has eigenvalues $\Lambda_1(s)$ with modulus $> 1$ and $R_{22}$ has eigenvalues $\Lambda_2(s)$ with modulus $< 1$, such that

$$A(s)Q(s) = Q(s)R(s)$$

(16)

and

$$Q^{-1}(0)Q(s) = \begin{bmatrix} U_{11}(s) & U_{12}(s) \\ U_{21}(s) & U_{22}(s) \end{bmatrix},$$

(17)

where the blocks $U_{11}$ and $U_{22}$ are symmetric positive definite (SPD).

Proof Suppose that $Q(s)$ and $R(s)$ satisfy (15) and (16). Let $Q'(s)$ and $R'(s)$ be any other pair that satisfies (15) and (16). Then we must have

$$Q'(s) = Q(s)T(s) = Q(s) \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix},$$

(18)

where $T$ is orthogonal, and also block diagonal.

Suppose also that in

$$Q^{-1}(0)Q'(s) = \begin{bmatrix} U_{11}' & U_{12}' \\ U_{21} & U_{22} \end{bmatrix}$$

(19)

both $U_{11}'$ and $U_{22}$ are SPD. By (18) and (19), we have

$$Q^{-1}(0)Q'(s) = Q^{-1}(0)Q(s) \begin{bmatrix} T_1 & 0 \\ 0 & T_2 \end{bmatrix} = \begin{bmatrix} U_{11}' & U_{12}' \\ U_{21} & U_{22} \end{bmatrix},$$

(20)
or, equivalently,

$$Q^{-1}(0)Q(s) = \begin{bmatrix} U_{11}'T_1^{-1} & U_{12}'T_2^{-1} \\ U_{21}'T_1^{-1} & U_{22}'T_2^{-1} \end{bmatrix}.$$
According to the polar decomposition of matrices [11], §4.2.10, $A_1$ and $A_2$, the upper left and lower right blocks of $Q^{-1}(0)Q(s)$ respectively, are uniquely presented as a product of an SPD matrix and an orthogonal matrix. This implies that $T_1$ and $T_2$ are uniquely defined.

Since $U(0) = I$, there is an open interval about 0, call it $I_0$, where we can require that $U_1$ has the structure

$$U_1(s) = \begin{bmatrix} I \\ U_{21}(s)U_{11}^{-1}(s) \end{bmatrix} U_{11}(s).$$

(22)

Next, for all $s \in I_0$, we define

$$Y_U(s) = U_{21}(s)U_{11}^{-1}(s).$$

(23)

The orthogonality relation $U_1^TU_1 = I$ implies:

$$U_1^T(s)U_1(s) = U_{11}^T(s)U_{11}(s) + U_{21}^T(s)U_{21}(s) = I.$$  

(24)

Using (24), we obtain

$$I + Y_U^TY_U = I + U_{11}^{-T}(s)U_{21}^T(s)U_{21}(s)U_{11}^{-1}(s)$$

$$= I + U_{11}^{-T}(s) \left[ I - U_{11}^T(s)U_{11}(s) \right] U_{11}^{-1}(s)$$

$$= I + U_{11}^{-T}(s)U_{11}^{-1}(s) - I$$

$$= U_{11}^{-T}(s)U_{11}^{-1}(s).$$

Now because $U_{11}$ is symmetric positive definite, $U_{11}^{-1}$ is the unique square root of $I + Y_U^TY_U$. This implies that we can rewrite (22) in terms of $Y_U$ and choose $U_{11}$ symmetric, to obtain

$$U_1 = \begin{bmatrix} I \\ Y_U \end{bmatrix} (I + Y_U^TY_U)^{-\frac{1}{2}}.$$  

(25)

Similarly, using $U_2^TU_2 = I$ and $U_1^TU_2 = 0$ for $U_2$, so that eventually we obtain for every $s \in I_0$,

$$U(s) = \begin{bmatrix} I \\ Y_U \end{bmatrix} \left( I + Y_U^TY_U \right)^{-\frac{1}{2}} \left( -\frac{Y_U^T}{I} \right) \left( I + Y_U^TY_U \right)^{-\frac{1}{2}}.$$

(26)
Hence, the columns of
\[
Q_U(0) \begin{bmatrix} I \\ Y_U \end{bmatrix}
\] (27)
form a base for the unstable eigenspace at \(x_1\) and the columns of
\[
Q_U^+(s) = Q_U(0) \begin{bmatrix} -Y_U^T \\ I \end{bmatrix}
\] (28)
form a base for the orthogonal complement of the unstable eigenspace. We note that these bases are in general not orthogonal.

Thus, we need to find the matrix \(Y_U \in \mathbb{R}^{(n-n_U) \times n_U}\) in (26). For any given \(s \in I_0\), define \(\hat{R}_{11}, \hat{R}_{12}, E_{21}\) and \(\hat{R}_{22}\) by
\[
Q^T(0)A(s)Q(0) = \begin{bmatrix} \hat{R}_{11} & \hat{R}_{12} \\ E_{21} & \hat{R}_{22} \end{bmatrix},
\] (29)
where \(\hat{R}_{11}\) is of size \(n_U \times n_U\) and \(\hat{R}_{22}\) is an \((n-n_U) \times (n-n_U)\) matrix.

By (11) and (12) we obtain the invariant subspace relation
\[
Q_2^+(s)A(s)Q_1(s) = 0.
\] (30)
Now we substitute \(Q(s)\) given by (13) and (26) and \(A(s)\) obtained from (29) into (30)
\[
[-Y_U \quad I]Q^T(0)Q(0) \begin{bmatrix} \hat{R}_{11} & \hat{R}_{12} \\ E_{21} & \hat{R}_{22} \end{bmatrix} Q^T(0)Q(0) \begin{bmatrix} I \\ Y_U \end{bmatrix} = 0
\] (31)
to obtain the following \textit{algebraic Riccati equation} for \(Y_U\):
\[
F(Y_U) = 0, \quad F(Y_U) := \hat{R}_{22}Y_U - Y_U\hat{R}_{11} + E_{21} - Y_U\hat{R}_{12}Y_U.
\] (32)

We now look at \(x_N\). In the same way we can compute a right invariant (stable) \(n_S\)-dimensional subspace \(S(\alpha)\) of \(A(\alpha)\). First, we consider \(Q(\alpha) = [Q_1(\alpha) \quad Q_2(\alpha)] \in \mathbb{R}^{n \times n}, Q_1(\alpha) \in \mathbb{R}^{n \times n_S}, Q_2(\alpha) \in \mathbb{R}^{n \times (n-n_S)}\) so that \(Q_1(\alpha)\) spans \(S(\alpha)\) and \(Q_2(\alpha)\) spans the orthogonal complement \(S^+(\alpha)\).

Using the same procedure as used in the computation of the unstable subspace for \(x_1\), we can obtain the relations
\[
Q(s) = Q(0)U(s), \quad \text{with} \quad U(0) = I,
\] (33)
and

\[ U(s) = \begin{bmatrix} I & Y_S^T Y_S \end{bmatrix} \begin{pmatrix} I + Y_S^T Y_S \end{pmatrix}^{-\frac{1}{2}} \begin{pmatrix} -Y_S^T \cr I \end{pmatrix} \begin{pmatrix} I + Y_S Y_S^T \end{pmatrix}^{-\frac{1}{2}}, \] (34)

and eventually the algebraic Riccati equation for \( Y_S \)

\[ F(Y_S) = 0, \quad F(Y_S) := \hat{R}_{22} Y_S - Y_S \hat{R}_{11} + E_{21} - Y_S \hat{R}_{12} Y_S \] (35)

to compute the stable invariant subspace and its orthogonal complement for \( x_N \). Solving (35) for \( Y_S \) of size \((n - n_S) \times n_S\), enables us to compute the span of the stable invariant subspace of \( x_N \) and its orthogonal complement. If \( Q_S(0) \) is the orthogonal matrix from the starting heteroclinic orbit, related to the stable invariant subspace, then a basis for the stable eigenspace in the new step at \( x_N \) is given by the columns of

\[ Q_S(0) \begin{bmatrix} I \\ Y_S \end{bmatrix}. \] (36)

A basis for the orthogonal complement of the subspace in the new step \( Q_S^\perp \), is given by the columns of

\[ Q_S^\perp(s) = Q_S(0) \begin{bmatrix} -Y_S^T \\ I \end{bmatrix}. \] (37)

These bases are in general not orthogonal.

2.2 Implementation

We now discuss the implementation of the algorithm in Cl_MATCONTM to continue the heteroclinic connection from the fixed point \( x_1 \) to the fixed point \( x_N \).

- **Continuation variables**
  The continuation variables are stored in a \( K \)-vector, where \( K = Nn + (n - n_U)n_U + (n - n_S)n_S + 1 \), containing:
  - A \( n \)-vector with the coordinates of the initial fixed point.
  - \( (N - 2) \) \( n \)-vectors with the coordinates of the mesh points \( x_2, \ldots, x_{N-1} \).
  - A \( n \)-vector \( x_N \) with the coordinates of the final fixed point.
  - The vector \( Y_U^r \), i.e. columnwise vectorized \( Y_U \).
  - The vector \( Y_S^r \), i.e. columnwise vectorized \( Y_S \).
  - An active parameter \( ap \), i.e. \( \alpha_a \).
• **Defining system**
  The defining systems consists of 
  \[ Nn + (n - n_U)n_U + (n - n_S)n_S \] 
  equations:
  • The initial fixed point constraint 
    \[ f^J(x_1, \alpha) - x_1 = 0. \]
  • The constraints 
    \[ f^J(x_{j-1}, \alpha) - x_j = 0, \quad j = 3, \ldots, N - 1. \]
  • The final fixed point constraint 
    \[ f^J(x_N, \alpha) - x_N = 0. \]
  • The rowwise vectorized Riccati equation (32) for \( Y_U \).
  • The rowwise vectorized Riccati equation (35) for \( Y_S \).
  • The initial boundary conditions (7).
  • The final boundary conditions (8).

• **Initialization**
  To implement the algorithm in CL\_MATCONT\_M [14], we need to initialize the connecting orbit, i.e. we set the problem parameter vector \( \alpha \), mesh points \( x_1, \ldots, x_N \), compute \( Q_1(0) \) and \( Q_2(0) \) corresponding to \( x_1 \) and \( x_N \) by (9) and initialize the vector \( Y_U = 0 \) and \( Y_S = 0 \) corresponding to the unstable and stable eigenspaces of \( x_1 \) and \( x_N \) of sizes \( ((n - n_U) \times n_U) \) and \( ((n - n_S) \times n_S) \), respectively. We also set a global structure \( \text{hets} \) containing the following fields:
  • Dimension of the state space (\( \text{hets.nphase} \)).
  • Number of mesh points, including the two fixed end points (\( \text{hets.npoints} \)).
  • The iteration number of the map \( J \) (\( \text{hets.niteration} \)).
  • Mapfile where the map is defined (\( \text{hets.mapfile} \)).
  • Vector of starting values of parameters and index of the active parameter (\( \text{hets.P0} \) and \( \text{hets.ActiveParams} \)).
  • Dimensions of the stable and unstable manifolds (\( \text{hets.nu} \) and \( \text{hets.ns} \)).
  • The matrices \( Q_U \) and \( Q_S \), bases for unstable and stable subspaces respectively (\( \text{hets.QU} \) and \( \text{hets.QS} \)).

• **Adaptation**
  At each continuation point a basis for the unstable eigenspace of \( x_1 \) is given by 
  \[ \text{hets.QU} \begin{bmatrix} I \\ Y_U \end{bmatrix} \]
  and for its orthogonal complement by 
  \[ \text{hets.QU} \begin{bmatrix} -Y_U^T \\ I \end{bmatrix}. \]
  However, these bases are not orthogonal. To restore orthogonality we must adapt \( Q_U \) from time to time. The base \( Q_U \) can be adapted using the singular value decomposition (SVD)
  \[ [U, S, V] = \text{svd} \left( \text{hets.QU} \begin{bmatrix} I \\ Y_U \end{bmatrix} \right), \]  
  \[ (38) \]
  where \( U \) and \( V \) are unitary matrices of sizes \( n \times n \) and \( n_U \times n_U \), respectively, and \( S \) is a diagonal matrix of size \( n \times n_U \). An adapted orthogonal base of the unstable subspace is given by \( U \). Then, the vector \( Y_U \) is set to zero.
By using a similar procedure we can adapt the matrices $Q_S$ and $Y_S$.

2.3 Computing the Jacobian matrix

To continue a connecting orbit, we need to compute the Jacobian matrix of the corresponding defining system. The Jacobian matrix can be computed by a finite difference approximation or by symbolic derivatives. However, continuation that uses finite differences is much slower in comparison with the symbolic Jacobian.

To compute the Jacobian matrix, we first initialize the Jacobian matrix $J$ as a zero sparse matrix of size $k \times (k+1)$, where $k = Nn + (n - n_U)n_U + (n - n_S)n_S$. We then compute the nonzero entries of $J$ by taking the derivatives of the system of defining equations with respect to the continuation variables.

- For the constraints in (4) we set $J(1:n, 1:n) = A(x_1) - I$ where $A(x_1)$ is the Jacobian of (1) at $x_1$ and $I$ is an identity matrix of size $n$. For derivatives of (4) w.r.t. the control parameter we set

$$J(1:n, k + 1) = (A_\alpha)_1(:, \text{hetds.ActiveParams}),$$

where $(A_\alpha)_1$ is the Jacobian of $f^J$ w.r.t. the parameter at $x_1$.

- For the $(N - 3)$ constraints defined in (5), we get for $j = 3, \ldots, N - 1$, $J((j-2)n+1 : (j-1)n, (j-2)n+1 : (j-1)n) = A(x_{j-1})$ and $J((j-2)n+1 : (j-1)n, (j-1)n+1 : jn) = -I$.

  For derivatives of (5) w.r.t. the control parameter we set

$$J((j-2)n+1 : (j-1)n, k + 1) = (A_\alpha)_j(:, \text{hetds.ActiveParams}),$$

where $j = 3, \ldots, N - 1$ and $(A_\alpha)_j$ is the Jacobian w.r.t. the control parameter at $x_j$.

- For the final fixed point constraint (6), $J$ is computed as: $J((N-2)n+1 : (N-1)n, (N-1)n+1 : Nn) = A(x_N) - I$. For derivatives of (6) w.r.t. the control parameter we set $J((N-2)n+1 : (N-1)n, k + 1) = (A_\alpha)_N(:, \text{hetds.ActiveParams})$, where $(A_\alpha)_N$ is the Jacobian of $f^J$ w.r.t. the parameter at $x_N$.

- Now we compute the entries of $J$ corresponding to (32) at $x_1$. First we consider the derivatives with respect to the components of $Y_U$. For simplicity of the computations we divide (32) into 3 terms $D_1 = \tilde{R}_{22}Y_U$, $D_2 = -Y_U\tilde{R}_{11}$, $D_3 = -Y_U\tilde{R}_{12}Y_U$.

- We start with $D_1$ whose derivatives with respect to the components of $Y_U$ are written into a block matrix of size $(n - n_U) \times n_U$ and $(n - n_U) \times n_U$. $D_1$ is an $(n - n_U) \times n_U$ matrix with general form $(D_1)_{(j,i)} = $
\[
\sum_{l=1}^{n-n_U} (\hat{R}_{22})_{(j,l)}(Y_U)_{(l,i)}, \quad j = 1, \ldots, n - n_U, \quad i = 1, \ldots, n_U. \]
Hence all nonzero derivatives arise from the fact that the derivative of \((D_1)_{(j,i)}\) with respect to \((Y_U)_{(s,i)}\) is \((\hat{R}_{22})_{(j,s)}\), \(0 \leq s \leq n - n_U\). Now if \(l = n(N - 1)\) and \(h = Nn\), then
- \((D_1)_{(j,i)}\) is at row position \(r_p = l + i + (j - 1)n_U\).
- \((Y_U)_{(s,i)}\) is at column position \(c_p = h + s + (i - 1)(n - n_U)\).

Therefore we set
\[
J(r_p, c_p) := (\hat{R}_{22})_{(j,s)},
\]
whenever \(1 \leq j \leq n - n_U, 1 \leq i \leq n_U\); and \(1 \leq s \leq n - n_U\).

- \(D_2\) is an \((n - n_U) \times n_U\) matrix with general form
\[
(D_2)_{(j,i)} = -\sum_{l=1}^{n_U} (Y_U)_{(j,l)}(\hat{R}_{11})_{(l,i)}, \quad j = 1, \ldots, n - n_U, \quad i = 1, \ldots, n_U.
\]
Hence all nonzero derivatives arise from the fact that the derivative of \((D_2)_{(j,i)}\) with respect to \((Y_U)_{(j,s)}\) is \((-\hat{R}_{11})_{(s,i)}\), \(1 \leq s \leq n_U\). Now if \(l = n(N - 1)\) and \(h = Nn\), then
- \((D_2)_{(j,i)}\) is at row position \(r_p = l + i + (j - 1)n_U\).
- \((Y_U)_{(j,s)}\) is at column position \(c_p = h + j + (s - 1)(n - n_U)\).

Therefore we update
\[
J(r_p, c_p) := J(r_p, c_p) - (\hat{R}_{11})_{(s,i)},
\]
whenever \(1 \leq j \leq n - n_U, 1 \leq i \leq n_U\); and \(1 \leq s \leq n_U\).

- \(D_3 = -Y_U \hat{R}_{12} Y_U\) is an \((n - n_U) \times n_U\) matrix. We introduce \(D_{31} = -Y_U \hat{R}_{12}\) and \(D_{32} = -\hat{R}_{12} Y_U\). With this notation, we have \(D_3' = D_{31} Y_U' + Y_U' D_{32}\) where the apostrophe denotes derivation with respect to an unspecified component of \(Y_U\).

First we consider \(D_3 Y_U'\) that is a \((n - n_U) \times n_U\) matrix with the general form \((D_3 Y_U')_{(j,i)} = \sum_{l=1}^{n-n_U} (D_{31})_{(j,l)}(Y_U')_{(l,i)}\). This contributes to the derivative with respect to \((Y_U)_{(r,s)}\) if \(i = s\) with the term \((D_{31})_{(j,r)}\). Now if \(l = n(N - 1)\) and \(h = Nn\), then
- \((D_{31} Y_U')_{(j,i)}\) is at row position \(r_p = l + i + (j - 1)n_U\);
- \((Y_U)_{(r,s)}\) is at column position \(c_p = h + r + (i - 1)(n - n_U)\).

Therefore we update \(J(r_p, c_p) := J(r_p, c_p) + (D_{31})_{(j,r)}\), whenever \(1 \leq j \leq n - n_U, 1 \leq i \leq n_U\); and \(1 \leq r \leq n - n_U\).

Now we consider \(Y_U D_{32}\) that is a \((n - n_U) \times n_U\) matrix with general form \((Y_U D_{32})_{(j,i)} = \sum_{l=1}^{n_u} (Y_U')_{(j,l)}(D_{32})_{(l,i)}\). This contributes to the derivative with respect to \((Y_U)_{(r,s)}\) if \(r = j\) with the term \((D_{32})_{(s,i)}\).
Now if \( l = n(N - 1) \) and \( h = Nn \), then

- \((Y_U' D_{32})(j,i)\) is at row position \( r_p = l + i + (j - 1)n_U\);
- \((Y_U)(j,s)\) is at column position \( c_p = h + j + (s - 1)(n - n_U)\).

Therefore we update

\[
J(r_p, c_p) := J(r_p, c_p) + (D_{32})(s,i),
\]

whenever \( 1 \leq j \leq n - n_U, 1 \leq i \leq n_U \) and \( 1 \leq s \leq n_U \).

We compute the derivatives of \( F(Y_U) \) in (32) with respect to the components of \( x_1 \). If \( 1 \leq i \leq n \), then the derivative of \( Q^T(0)A(s)Q(0) \) w.r.t. \( x_{(i,1)} \) is given by \( D_i = Q^T(0)hess(:,i)Q(0) \) where \( hess \) is the Hessian of \( f^J \). Then \( D_{1i} = D_i(1 : n_U, 1 : n_U), D_{2i} = D_i(1 : n_U, n_U + 1 : n), D_{3i} = D_i(n_U + 1 : n, 1 : n_U) \) and \( D_{4i} = D_i(n_U + 1 : n, n_U + 1 : n) \) are derivatives of \( \dot{R}_{11}, \dot{R}_{12}, E_{21} \) and \( \dot{R}_{22} \) w.r.t. \( x_{(1,i)} \), respectively. Derivatives of \( F(Y_U) \) w.r.t. \( x_{(1,i)} \) are hence given by

\[
(F(Y_U))_{x_{(1,i)}} = D_{1i}Y_U - Y_U D_{2i} + D_{3i} - Y_U D_{4i}Y_U.
\]

All nonzero derivatives arise from the fact that the derivative of \( (F(Y_U))(j,s) \) w.r.t. to \( x_{(1,i)} \) is

\[
(D_{1i})(j,:) (Y_U)_{(:,s)} - (Y_U)(j,:) (D_{2i})_{(:,s)} + (D_{3i})(j,:) - (Y_U)(j,:) (D_{4i}Y_U)_{(:,s)}.
\]

Now if \( l = n(N - 1) \), then

- \((F(Y_U))(j,s)\) is at row position \( r_p = l + s + (j - 1)n_U\);
- \( x_{(1,i)} \) is at column position \( c_p = i \).

Therefore we set

\[
J(r_p, c_p) := (D_{1i})(j,:) (Y_U)_{(:,s)} -
\]

\[
(Y_U)(j,:) (D_{2i})_{(:,s)} + (D_{3i})(j,:) - (Y_U)(j,:) (D_{4i}Y_U)_{(:,s)},
\]

whenever where \( 1 \leq j \leq n - n_U, 1 \leq s \leq n_U, 1 \leq i \leq n \).

We now compute the derivatives of \( F(Y_U) \) in (32) with respect to the control parameter \( \alpha_a \). The derivative of \( Q^T(0)A(s)Q(0) \) w.r.t. the control parameter is given by \( D = Q^T(0)hessp(:,i,\text{ActiveParams})Q(0) \) where \( hessp \) is the Hessian of \( f^J \) w.r.t. \( \alpha_a \). Then \( D_1 = D(1 : n_U, 1 : n_U), D_2 = D(1 : n_U, n_U + 1 : n), D_3 = D(n_U + 1 : n, 1 : n_U) \) and \( D_4 = D(n_U + 1 : n, n_U + 1 : n) \) are derivatives of \( \dot{R}_{11}, \dot{R}_{12}, E_{21} \) and \( \dot{R}_{22} \), w.r.t. \( \alpha_a \), respectively.

Derivatives of \( F(Y_U) \) w.r.t. the control parameter are hence given by

\[
(F(Y_U))_{\alpha_a} = D_1Y_U - Y_U D_2 + D_3 - Y_U D_4Y_U.
\]
All nonzero derivatives arise from the fact that the derivative of \((F(Y_U))_{(j,s)}\) w.r.t. to \(\alpha_a\) is

\[
(D_1)_{(j,:)}(Y_U)_{(:,s)} - (Y_U)_{(j,:)}(D_2)_{(:,s)} + (D_3)_{(j,s)} - (Y_U)_{(j,:)}(D_4 Y_U)_{(:,s)}.
\]

Now if \(l = n(N-1)\), then
- \((F(Y_U))_{(j,s)}\) is at row position \(r_p = l + s + (j-1)n_U\);
- \(\alpha_a\) is at column position \(c_p = k + 1\).

Therefore we set

\[
J(r_p, c_p) := (D_1)_{(j,:)}(Y_U)_{(:,s)} - (Y_U)_{(j,:)}(D_2)_{(:,s)} + (D_3)_{(j,s)} - (Y_U)_{(j,:)}(D_4 Y_U)_{(:,s)},
\]

whenever where \(j = 1, \ldots, n - n_U\), \(s = 1, \ldots, n_U\).

The derivatives that correspond to (35) can be obtained similarly as those of (32).

- We consider conditions (7) which consists of \((n - n_U)\) equations of the form

\[
B_i = \sum_{k=1}^{n} (x_2 - x_1)_{(1,k)}(q^U_{(n_U+i)})_k = 0, \quad i = 1, \ldots, n - n_U.
\]

- All nonzero derivatives of \(B_i\) w.r.t. \(x_{(1,s)}\) are of the form \(-q^U_{(n_U+i)}\). Now if \(l = n(N-1) + n_U(n - n_U) + n_S(n - n_S)\), then
  - \(B_{(1,i)}\) is at row position \(l + i\);
  - \(x_{(1,s)}\) is at column position \(s\).

Therefore we set

\[
J(l + i, s) := -(q^U_{(n_U+i)})_s,
\]

whenever \(1 \leq i \leq n - n_U\) and \(1 \leq s \leq n\).

- All nonzero derivatives of \(B_i\) w.r.t. \(x_{(2,s)}\) are of the form \((q^U_{(n_U+i)})(s)\). Now if \(l = n(N-1) + n_U(n - n_U) + n_S(n - n_S)\), then
  - \(B_{i}\) is at row position \(l + i\);
  - \(x_{(2,s)}\) is at column position \(n + s\).

Therefore we set

\[
J(l + i, n + s) := (q^U_{(n_U+i)})_s,
\]

whenever \(1 \leq i \leq n - n_U\) and \(1 \leq s \leq n\).
We now compute derivatives of (7) w.r.t. the components of $Q_U$. The equations (7) have the following form

$$B_i = (x_2 - x_1)^T q_{nU+i}^U = 0, \text{ for } i = 1, \ldots, n - n_U. \quad (39)$$

Here $q_{nU+i}^U$ are precisely the columns of $Q_U(0) \left[ -Y_U^T I \right]$. Therefore it is best to introduce the new vector $H = (x_2 - x_1)^T \hat{Q}(0)$. With this notation the $i$-th equation becomes

$$- \sum_{j=1}^{n_U} H_j (Y_U^T)_{ji} \text{ + terms without components of } Y_U = 0.$$

This means that the $i$-th equation has derivatives with respect to $(Y_U)_{ji}$ equal to $-H_j \ (i = 1, \ldots, n - n_U, \ j = 1, \ldots, n_U)$. Let $l = n(N-1) + n_U(n - n_U) + n_S(n - n_S), \ h = nN$. Then the $i$-th equation is at column position $l + i$. The variable $(Y_U)_{ji}$ is at column position $h + j + (n - n_U)(i - 1)$. So we have to set

$$J(l + i, \ h + j + (n - n_U)(i - 1)) := -H_j, \ \text{ for the relevant } i, j.$$

The derivatives that corresponds to (8) can be computed as those that correspond to (7).

3 Continuation of homoclinic connections

Assume that the eigenvalues of $(f^J(x_1, \alpha))_x$ are ordered as follows:

$$|\lambda_1| \leq \ldots \leq |\lambda_m| < 1 < |\lambda_{m+1}| \leq \ldots \leq |\lambda_n|.$$ 

The procedure to continue a homoclinic connection to $x_1$ is similar to the procedure used in Section 2. The algorithm now requires the evaluation of two projections associated with the eigenspaces of $(f^J(x_1, \alpha))_x$. These projections are constructed using the real Schur factorizations

$$(f^J(x_1, \alpha))_x = Q^{(1)} R^{(1)} [Q^{(1)}]^T, \quad (f^J(x_1, \alpha))_x = Q^{(N)} R^{(N)} [Q^{(N)}]^T,$$

where $Q^{(1)}, R^{(1)}, Q^{(N)},$ and $R^{(N)}$ are $n \times n$-matrices. The first factorization has been chosen so that the first $m$ columns $q_1^S, \ldots, q_m^S$ of $Q^{(1)}$ form an orthonormal basis of the right invariant subspace $S_1$ of $(f^J(x_1, \alpha))_x$, corresponding to
\(\lambda_1, \ldots, \lambda_m\) and the remaining \(n - m\) columns \(q_{m+1}^U, \ldots, q_n^U\) of \(Q^{(1)}\) form an orthonormal basis of \(S_1^+\). Similarly, the first \(l = n - m\) columns \(q_1^U, \ldots, q_l^U\) of \(Q^{(N)}\) form an orthonormal basis of the right invariant subspace \(U_1\) of \((f^J(x_1, \alpha))_x\), corresponding to \(\lambda_{m+1}, \ldots, \lambda_n\) and the remaining \(n - l - m\) columns \(q_{l+1}^U, \ldots, q_n^U\) of \(Q^{(1)}\) form an orthonormal basis of \(S_N^+\).

When dealing with homoclinic connections, we want to find a sequence of points \((x_k)_{k=1, \ldots, N-1}\) satisfying

- Stationary state condition
  \[
  f^J(x_1, \alpha) - x_1 = 0; \tag{40}
  \]
- The iteration conditions
  \[
  f^J(x_k, \alpha) - x_{k+1} = 0, \quad k = 2, 3, \ldots, N - 2; \tag{41}
  \]
- The left projection boundary conditions
  \[
  (x_2 - x_1)^T q_{n_U+i}^U(\alpha) = 0, \quad i = 1, \ldots, n - n_U; \tag{42}
  \]
- The right projection boundary conditions
  \[
  (x_{N-1} - x_1)^T q_{n_S+i}^S(\alpha) = 0, \quad i = 1, \ldots, n - n_S. \tag{43}
  \]

A regular zero of a system of equations (40), (41), (42) and (43) corresponds to a transversal homoclinic orbit to a hyperbolic fixed point. Thus, a zero for this system can be continued in one parameter.

The equations (42) and (43) imply that we should have the stable and unstable eigenspaces of the map (1) at the fixed points \(x_1\) available at each step of the continuation. Using the same procedure, as in the computation of the unstable subspace for \(x_1\), we can obtain relations analogous to (13), (26) and (32) to compute the stable invariant subspace and its orthogonal complement at \(x_1\).

### 4 Invariant manifolds of planar maps

Our main motivation for computing stable and unstable manifolds of a saddle point is the role they play in the computation of connecting orbits. Intersections of stable and unstable manifolds give rise to homoclinic or heteroclinic tangles. Stable and unstable manifolds are global objects that cannot be found analytically and, hence, must be computed numerically. These manifolds must
be grown from local knowledge, for example from linear information near a fixed point.

We concentrate here on the simplest case of planar maps when these manifolds are one-dimensional. Most algorithms use the idea of computing the manifold by starting from a local approximation near the saddle point. The map that arises in a particular application does not necessarily have an explicit inverse or may not even be invertible, meaning that there may be several branches of inverses. Consequently, the standard algorithms requiring the inverse cannot be used to compute stable manifolds of saddle points in this case.

First we present a known algorithm to compute the stable manifold of a saddle point of a planar map, without requiring any knowledge of its inverse map, either explicitly or approximately. In particular, the algorithm can also be used in the case where the map is noninvertible, so that multiple pre-images may exist. Then we focus on the computation of the unstable manifolds.

We recall some definitions, mostly to fix the notation. We consider (1) when \( n = 2 \) and assume that \( f \) has a fixed point \( x_0 = f^J(x_0) \) and that \( f \) is differentiable in a neighborhood of \( x_0 \), but may not have a single-valued inverse. The fixed point \( x_0 \) of \( f \) is a saddle if the Jacobian matrix \( (f^J(x_0))_x \) has one stable eigenvalue \( \lambda_s \) and one unstable eigenvalue \( \lambda_u \). The stable manifold theorem [26] guarantees that there exist local stable and unstable manifolds \( W_s^{\text{loc}}(x_0) \) and \( W_u^{\text{loc}}(x_0) \) tangent at \( x_0 \) to the stable and unstable eigenspaces \( E^s(x_0) \) and \( E^u(x_0) \), respectively. The global stable manifold \( W_s(x_0) \) of \( x_0 \) is defined as the set of points that converge to \( x_0 \) under forward iteration of \( f \),

\[
W_s(x_0) = \left\{ x \in \mathbb{R}^2 : f^J(x) \to x_0 \text{ as } J \to \infty \right\}. \tag{44}
\]

Similarly, the global unstable manifold \( W_u(x_0) \) of \( x_0 \) consists of points that converge to \( x_0 \) under backward iteration of the invertible map \( f \). In terms of forward iterates, this is defined as

\[
W_u(x_0) = \left\{ x \in \mathbb{R}^2 : \exists \{q_k\}, q_0 = x \text{ and } f^J(q_{k+1}) = q_k, \text{ and } \lim_{k \to \infty} q_k = x_0 \right\}. \tag{45}
\]

The stable manifold \( W_s(x_0) \) is often defined as the union of the successive pre-images of \( W_s^{\text{loc}}(x_0) \). However, if multiple inverses exist, then all pre-images, even if disjoint from the main branch, are part of the stable manifold. Hence, thus defined, the stable manifold may or may not be simply connected in phase space.
4.1 Computing a stable manifold

To compute the one-dimensional stable manifold of a saddle point of a planar map, we use the Search Circle (SC) algorithm proposed in [8]. We assume that \( f \) is orientation preserving; otherwise we consider its second iterate. We briefly explain this algorithm that extends the one-dimensional manifold in steps by adding new points according to the local curvature properties of the manifold, and finds a new point close to the last computed point that maps under \( f \) to a piece of the manifold that was already computed. SC produces a piecewise linear approximation \( W_{pl}^s \) of \( W^s(x_0) \) by computing an ordered list of points \( M = \{p_0, p_1, \ldots, p_n\} \) at varying distance from each other. The first point \( p_1 \) is taken a small distance \( \delta > 0 \) from \( p_0 = x_0 \) along \( E^s(x_0) \). The distance between consecutive points is adjusted according to the curvature of the manifold. To ensure an acceptable resolution of the curve according to pre-specified accuracy parameters, we monitor \( \alpha_k \), the angle between \( p_k \) and \( p_{k+1} \), and the product \( \alpha_k \Delta_k \). The \( \alpha_k \) is approximated by

\[
\alpha_k = 2 \sin^{-1} \left( \frac{\| \tilde{p} - p_{k-1} \|}{2 \| p_k - p_{k-1} \|} \right) \approx \frac{\| \tilde{p} - p_{k-1} \|}{\| p_k - p_{k-1} \|}, \tag{46}
\]

where

\[
\tilde{p} = p_k + \frac{\| p_k - p_{k-1} \|}{\| p_k - p_{k+1} \|} (p_k - p_{k+1}) \tag{47}
\]

is the point on the line through \( p_k \) and \( p_{k+1} \) that lies at the same distance from \( p_k \) as \( p_{k-1} \). We check the conditions

\[
\alpha_k < \alpha_{\text{max}} \tag{48}
\]

\[
\alpha_k \Delta_k < (\alpha \Delta)_{\text{max}} \tag{49}
\]

Condition (48) ensures that the resolution of the curve is maintained and condition (49) controls the local interpolation error. The new point \( p_{k+1} \) is accepted if it satisfies the above criteria. If one of the criteria is not satisfied, then we replace \( \Delta_k \) by \( \frac{\Delta_k}{2} \) and repeat the procedure to find a new candidate for \( p_{k+1} \). We set \( \Delta_{k+1} = 2 \Delta_k \) if both \( \alpha_k > \alpha_{\text{min}} \) and \( \alpha_k \Delta_k > (\alpha \Delta)_{\text{min}} \) for a user-specified choice of parameters \( \alpha_{\text{min}} \) and \( (\alpha \Delta)_{\text{min}} \). This ensures that the number of points used to approximate the manifold is in some sense optimized for the required accuracy constraints, see [21] for more details.
Figure 1. A graphical illustration of the SC algorithm. A new point $p_{k+1}$ is found on the circle $C(p_k, \Delta_k)$ centered at $p_k$ with radius $\Delta_k$ such that $f(p_{k+1})$ lies on a previously computed part of $W^S(x_0)$.

A pseudo-code representation of how the branch is grown can be found in [8]. A graphical illustration of the SC algorithm is given in Figure 1.

4.2 Computing an unstable manifold

We use the algorithm for computing the global one-dimensional unstable manifold of a saddle point of a map proposed in [20]. To keep the exposition simple, we consider a planar diffeomorphism and suppose $f$ is orientation-preserving, otherwise we consider its second iterate. Let $x_0$ be a saddle point of $f$. The unstable manifold of $x_0$ can now be defined as

$$W^u(x_0) = \{ x \in \mathbb{R}^2 : f^{-J}(x) \to x_0 \text{ as } J \to \infty \}.$$  

(50)

now Note that, since $f$ is invertible, the stable manifold $W^s(x_0)$ is simply the unstable manifold of $f^{-1}$ at $x_0$. The unstable manifold theorem [26] guarantees the existence of the local unstable manifold

$$W^u_{loc}(x_0) = \{ x \in W^u(x_0) : f^{-n}(x) \in U \text{ for all } n \in \mathbb{N} \}.$$  

(51)

in a suitable neighborhood $U$ of $x_0$. Furthermore, it states that $W^u_{loc}(x_0)$ is tangent to the unstable eigenspace $E^u(x_0)$ corresponding to $\lambda^U$.

Similar to the algorithm that is used for computing the stable manifold, the idea is to grow the manifold independently of the dynamics in steps as a list of ordered points. At each step a new point is added at a prescribed distance $\Delta_k$ from the last point. New points are found as $f$-images of suitable points from the part already computed. The algorithm starts with a linear approximation of the local manifold and grows the manifold up to a prespecified arclength $l$ with a speed depending on the local curvature of the manifold.

We now briefly describe a single step of the algorithm and suppose that the piecewise-linear approximation $W^u_{pl}$ of the manifold defined by points $M =
{p_0, p_1, \ldots, p_k} is already computed, where p_0 = x_0 and the point p_1 is taken at a small distance δ from x_0 in the unstable eigenspace E^u(x_0). The next point p_{k+1} should have the property that the line segment [p_k, p_{k+1}] accurately approximates W^u(x_0). In order to achieve a good approximation, the distance δ between p_k and p_{k+1} must be adjusted from step to step according to the curvature of the manifold.

We want to find p_{k+1} in a small annulus around the circle at p_k with radius \Delta_k. To this end, we search in W^u_{pl}(x_0) a line L that is mapped by f to a curve which intersects the circle with center p_k and radius \Delta_k. We start with the line segment in W^u_{pl}(x_0) that contains the preimage of p_k and move linearly through W^u_{pl}(x_0). Once L is found, we use bisection to find a point q \in L such that

\[(1 - \epsilon)\Delta_k < \|f(q) - p_k\| < (1 + \epsilon)\Delta_k\]

The point p_{k+1} = f(q) is a candidate for the next point in M, see Figure 2. If \Delta_k is acceptable, then p_{k+1} = f(q) is added to M, [p_k, p_{k+1}] is added to W^u_{pl}(x_0), and the step is completed. However, if \Delta_k was too large then we reject f(q), half the estimate \Delta_k, and repeat the procedure. This algorithm is presented in the pseudo-code in [21].

5 Continuation of heteroclinic and homoclinic tangencies

Let F(X, \alpha) = 0 be the defining system of the heteroclinic connection introduced in Section 2.2, then a heteroclinic tangency satisfies the following
limit-point conditions:

\[
\begin{aligned}
F(X, \alpha) &= 0, \\
\det(F_X(X, \alpha)) &= 0,
\end{aligned}
\tag{52}
\]

which is a system of \( K_1 = n(N - 1) + n_U(n - n_U) + n_S(n - n_S) + 2n - n_U - n_S + 1 \) equations in an \( K_2 = (nN + n_U(n - n_U) + n_S(n - n_S) + n_\alpha) \)-dimensional space. We recall that \( X = (x_1, \ldots, x_N, Y_U, Y_S, \alpha) \).

If \( n_U + n_S = n \) and \( n_\alpha = 2 \), then (52) defines a generic continuation problem. This system is natural from a theoretical perspective but may lead to numerical scaling problems. If the Jacobian has eigenvalues of large magnitude, then these eigenvalues contribute to the determinant (which is the product of all eigenvalues) and may make it difficult to satisfy the defining equations to a desired tolerance. The larger the system, the worse this problem becomes. Thus there are circumstances in which it is desirable to seek alternate defining equations that avoid calculation of the determinant. Bordered matrices allow us to find a substitute function of the determinant.

We define a curve of heteroclinic tangencies by the following system

\[
\begin{aligned}
F(X, \alpha) &= 0, \\
g(X, \alpha) &= 0,
\end{aligned}
\tag{53}
\]

where \( g(X, \alpha) \) is computed as the last component of the solution vector in the \( K_1 \)-dimensional bordered system:

\[
\begin{pmatrix}
F_X(X, \alpha) \\
c^T
\end{pmatrix}
\begin{pmatrix}
b \\
g
\end{pmatrix}
= \begin{pmatrix}
0_{(K_1 - 1)} \\
1
\end{pmatrix},
\tag{54}
\]

for suitable vectors \( b, c \in \mathbb{R}^{K_1 - 1} \).

If \( c \) is close to the nullvector of \( F_X(X, \alpha) \) and \( b \) is close to the nullvector of \( F_X^T(X, \alpha) \), then the matrix

\[
M = \begin{pmatrix}
F_X(X, \alpha) \\
c^T
\end{pmatrix}
\begin{pmatrix}
b \\
g
\end{pmatrix}
\tag{55}
\]

is nonsingular at \((X, \alpha)\) and (54) has a unique solution. In practical computations, \( c \) and \( b \) are approximations of the null vectors of \( F_X(X, \alpha) \) and \( F_X^T(X, \alpha) \), respectively.

In the continuation of heteroclinic tangencies \( b \) and \( c \) are computed in the curve initializer \texttt{init.HetT.HetT} and stored in the fields \texttt{hetTds.b} and \texttt{hetTds.c} of the global variable \texttt{hetTds}.
The vectors \( b \) and \( c \) must be adapted during the continuation of heteroclinic tangencies to keep the matrix \( M \) nonsingular. Towards this end we use the SVD decomposition \( [U,S,V] = \text{svd}(\text{full}(F_X(X,\alpha))) \) where \( U,V \) are orthogonal matrices and \( S \) is a diagonal matrix, and \( F_X(X,\alpha) = USV^T \). Using the fact that \( c \) is a normalized right nullvector of \( F_X \) we have:

\[
F_X(X,\alpha)c = USV^Tc = 0.
\]

By the orthogonality of \( U \), we get \( SV^Tc = 0 \). The only possibility for the null vector of \( S \) is \( V^Tc = [0,\ldots,0,1]^T \). Since \( V \) is an orthogonal matrix, we finally obtain: \( c = V[0,\ldots,0,1]^T \). That means \( c \) is the last column of \( V \).

For the left nullvector \( b \), we have

\[
b^TF_X(X,\alpha) = b^TUSV^T = 0.
\]

By the orthogonality of \( V \), we get \( b^TUS = 0 \). The only possibility for the null vector of \( S \) is \( b^TU = [0,\ldots,0,1] \). Since \( U \) is an orthogonal matrix, we finally obtain: \( b = U[0,\ldots,0,1]^T \). That means \( b \) is the last column of \( U \).

By now it is fairly clear that homoclinic tangencies can be computed in essential the same way.

6 Examples and applications

We consider the generalized Hénon map (GHM)

\[
F : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} \alpha - \beta x_1 - x_2^2 + Rx_1x_2 + Sx_2^3 \end{pmatrix},
\]

which appears in numerous theoretical studies of homoclinic bifurcations, see [12]. In what follows we set \( S = 0 \).

6.1 Heteroclinic connection

For \( \alpha = 0.3, \beta = -1.057, \) and \( R = -0.5 \), \( F \) has two fixed points, namely

\[
X_0 = (0.4666170238049, 0.4666170238049),
\]

\[
X_1 = (-0.4286170238049, -0.4286170238049),
\]

with multipliers \(-1.662111803, .4955692430\) at \( X_0 \) and multipliers \( 1.784113932, -.7125713719 \) at \( X_1 \). Since \( F \) is now orientation-reversing, we
will consider the second iterate $F^2$ to compute heteroclinic connections and, in particular, $\lambda_2^U = 2.762615646$.

We use the algorithm for computing the unstable manifold as described in Section 4.2, to construct the one-dimensional manifold of the saddle point $X_0$, by calling the newly implemented MATLAB function $Umanifold.m$. To this end, first we set $\text{amax}$, $\text{amin}$, $\text{dmax}$, $\text{dmin}$, $\text{damax}$, $\text{damin}$, $\text{dk}$ and $\text{epsb}$ corresponding to $\alpha_{\text{max}}$, $\alpha_{\text{min}}$, $\Delta_{\text{max}}$, $\Delta_{\text{min}}$, $(\Delta \alpha)_{\text{max}}$, $(\Delta \alpha)_{\text{min}}$, $\Delta_k$ and $\epsilon_b$, respectively, as described in §4.2. The input commands are:

```matlab
epsb=1e-10; Arc=8.6; dk=1e-3; amax=0.6; amin=0.2;
dmin=0.001; damax=0.07; damin=1e-4; dmax=0.2;
p0=[0.46661702380495;0.46661702380495];
p=[0.3;-1.057;-0.5;0];%alpha,beta,R,S
lambda=2.76261564458262; del=1e-2;
v=[0.51553133957551;-0.85687072415592];
M1=Umanifold(p0,Arc,amax,amin,dmax,dmin,damax,damin,
dk,p,epsb,lambda,del,v);
for i=1:size(M1,2) hold on, plot(M1(1,i),M1(2,i),'r.'), end
del=-1e-4;
M2=Umanifold(p0,Arc,amax,amin,dmax,dmin,damax,damin,
dk,p,epsb,lambda,del,v);
for i=1:size(M2,2) hold on, plot(M2(1,i),M2(2,i),'r.'), end
plot(0.46661702380495,0.46661702380495,'--sk')
```

where $p_0$ represents the saddle point $X_0$, $p$ is the vector of parameters $(\alpha, \beta, R, S)$, $\lambda$ is the unstable multiplier of $F_x$ at $X_0$ and $\text{Arc}$ is the total length of the unstable manifold to be computed. The plot of the two unstable manifolds was given in Figure 3. The elapsed time to compute 215200 points was 1470 seconds on a AMD Athlon XP 2800+ processor. We note that a continuation run to compute 100 period-2 cycles of $F$ takes about 2.4 seconds.

We now use the algorithm for computing the stable manifold as described in Section 4.1, to obtain the stable manifold of $X_1$. The commands are:

```matlab
epsb=1e-7; Arc=5; dk=1e-4; amax=0.5; amin=0.2;
dmin=0.01; damax=0.7; damin=1e-3 ;dmax=1e-2;
p0=[-0.42861702380495;-0.42861702380495];
p=[0.3;-1.057;-0.5;0];%alpha,beta,R,S
p1=p0-1e-3*[0.8143932867458;0.58031334165721];%p0-1e-3*v
M1=Smanifold(p0,p1,Arc,amax,amin,dmax,dmin,damax,damin,
dk,p,epsb,n);
for i=1:size(M1,2) hold on, plot(M1(1,i),M1(2,i),'k.'), end
```

where $p_0$ represents the saddle point $X_0$, $p$ is the vector of parameters $(\alpha, \beta, R, S)$, $\lambda$ is the unstable multiplier of $F_x$ at $X_0$ and $\text{Arc}$ is the total length of the unstable manifold to be computed. The plot of the two unstable manifolds was given in Figure 3. The elapsed time to compute 215200 points was 1470 seconds on a AMD Athlon XP 2800+ processor. We note that a continuation run to compute 100 period-2 cycles of $F$ takes about 2.4 seconds.
Figure 3. The transversal intersection of unstable and stable manifolds of $F$ at $X_0$ and $X_1$, respectively.

$p1=p0+1e-3*[-0.81439328674458;0.58031334165721];%p0+1e-3*v$
\[ M2=Smanifold(p0,p1,Arc,amax,amin,dmax,dmin,damax,damin, dk,p,epsb,n); \]
\[ \text{for } i=1:\text{size}(M2,2) \text{ hold on, plot}(M2(1,i),M2(2,i),'k.'), \text{ end} \]
\[ \text{plot}(-0.42861702380495;-0.42861702380495,'--sk') \]

The plot of the two stable manifolds is given in Figure 3. The elapsed time to compute 92000 points was 560 seconds.

We will now continue w.r.t. parameter $\beta$ the heteroclinic orbit in Cl_\text{MATCONT}, using the transversal intersection of invariant manifolds at $X_0$ and $X_1$, as an initial approximation. The set of intersection points \{x_5,x_6,\ldots,x_{10}\} from Figure 4 is an initial approximation. To get a more accurate heteroclinic orbit we extend it by adding more points. To this end, we use iterations of $F^{-1}$ and $F$ and project the resulting points on $E^u(x_0)$ and $E^s(x_1)$, respectively.

We start from $x_5$ and compute the point $F^{-1}(x_5)$. By projecting the resulting point on $E^u(x_0)$, we compute $x_4$ as an approximation of a new intersection point of $W^u(x_0)$ and $W^s(x_1)$. We then apply $F^{-1}$ on $x_4$ and by projecting the new point on $E^u(x_0)$ we compute $x_3$. We proceed with the same steps to compute the points $x_2$ and $x_1$.

We now use the same procedure by applying $F$ on $x_{10}$. By projecting the resulting point on $E^s(x_1)$ we compute $x_{11}$ as an approximation of an intersec-
Figure 4. The transversal intersection of the unstable and stable manifolds of $F$ at $X_0$ and $X_1$, respectively.

The transversal intersection of $W^u(x_0)$ and $W^s(x_1)$. We repeat the same steps to compute the points $x_{12}, x_{13}, \ldots, x_{16}$.

The resulting initial approximation is given by

$$C = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}\},$$

where

$$x_1 = (0.4666171; 0.4666171), \quad x_2 = (0.519502; 0.376394),$$
$$x_3 = (0.483172; 0.439158), \quad x_4 = (0.4731169; 0.456962),$$
$$x_5 = (0.612300; 0.206700), \quad x_6 = (0.841195; -0.276064),$$
$$x_7 = (1.229904; -1.332700), \quad x_8 = (0.641982; -1.093020),$$
$$x_9 = (0.134731; -0.799843), \quad x_{10} = (-0.143457; -0.623386),$$
$$x_{11} = (-0.333799; -0.495162), \quad x_{12} = (-0.386201; -0.462550),$$
$$x_{13} = (-0.404242; -0.445916), \quad x_{14} = (-0.416213; -0.437437),$$
$$x_{15} = (-0.422303; -0.433111), \quad x_{16} = (-0.428617; -0.428617).$$

The code below is the implementation into Cl_MATCONT:

```matlab
C={x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16}
p=[0.3;-1.957;0.5;0]; ap=[2]; opt = contset;
[x0,v0]=init_HE_HE(@Ghmap,C, p, ap,2);
opt=contset(opt,'MaxNumpoints',30);
opt=contset(opt,'Singularities',1);
```
Connecting orbits of maps

```
Figure 5. Tangential intersection of the invariant manifolds of (56) for 
\( \alpha = 0.3; \beta = -1.009322; R = -0.5; S = 0 \)
```

```
opt=contset(opt,'Backward',1);
[xhet,vhet,shet,hhet,fhet]=cont(@heteroclinic,x0,[]},opt);
```

We detect two limit points (LP) on the heteroclinic orbit:

```
label = LP , x = ( 0.450332 0.450332 0.464235 ... -1.009322)
label = LP , x = ( 0.471227 0.471227 0.487755 ... -1.070206)
elapsed time = 1.8 secs
npoints curve = 30
```

In the computed LP points the first 32 components indicate the coordinates of the mesh points \( x_1, \ldots, x_{16} \), the following 2 (not shown) indicate \( Y_U \) and \( Y_S \) in the Riccati equations (32) and (35), respectively, and the last shown component is the value of the control parameter \( \beta \). A picture of the computed branch of heteroclinic orbits is given in Figure 6.

For the parameter values of the fold points, i.e., \( \alpha = 0.3, \beta = -1.009322, R = -0.5 \), and \( \alpha = 0.3, \beta = -1.070206, R = -0.5 \), we have a tangential intersection of the invariant manifolds. A tangential intersection of invariant manifolds is shown in Figure 5.

Next we continue the limit points in two parameters, starting from the LP on the heteroclinic connections. This curve is shown in Figure 7. The elapsed time to compute 40 points was about 109 seconds.

### 6.2 Homoclinic connection

Now we consider the parameter values \( \alpha = -0.4, \beta = 1.03, \) and \( R = -0.1 \). Here \( F \) preserves orientation and has a saddle fixed point \( X_0 = (-1.62114638486, -1.62114638486) \) with the multipliers 0.2775591559 and
We compute $W^u(X_0)$ and $W^s(X_0)$ and determine their intersection points to be used as initial data for the homoclinic continuation. Figure 8 depicts $W^u(X_0)$ and $W^s(X_0)$ along with their intersection points. We continue the homoclinic orbit w.r.t. parameter $\beta$ in Cl_MATCONT, using the transversal intersection of invariant manifolds at $X_0$, as an initial approximation. The initial approximation is given by

$$C = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\},$$
where
\[
\begin{align*}
x_1 &= (-1.62114638; -1.62114638), \quad x_2 = (-1.56200000; -1.44300000), \\
x_3 &= (-1.44300000; -1.09878560), \quad x_4 = (-1.09878560; -0.27959456), \\
x_5 &= (-0.27959456; 0.62285460), \quad x_6 = (0.62285460; -0.48255079), \\
x_7 &= (-0.48255079; -1.24433961), \quad x_8 = (-1.24433961; -1.51139945), \\
x_9 &= (-1.51139945; -1.59072793), \quad x_{10} = (-1.59072793; -1.61409646).
\end{align*}
\]

In Figure 9, obtained by simulations, we present a destructed closed invariant curve for \(\alpha = -0.4, \beta = 1.03, R = -0.1\), superposed on Figure 8. This invariant curve is born via the Neimark-Sacker bifurcation and is destructed by approaching the homoclinic tangle.

We detect two limit points (LP):
\[
\begin{align*}
x &= ( -1.704631 -1.704631 -1.668284 \ldots 1.109749 ) \\
x &= ( -1.586188 -1.586188 -1.559729 \ldots 0.996984 )
\end{align*}
\]

elapsed time = 1.3 secs
npoints curve = 35

In the computed LP points the first 20 components indicate the coordinates of the mesh points \(x_1, \ldots, x_{10}\), the following 2 indicate \(Y_U\) and \(Y_S\) in the Riccati equations (32) and (35), respectively and the last component is the value of the control parameter \(\beta\). The computed branch of homoclinic connections is presented in Figure 10.

Now we can continue the curve of limit points in two parameters, starting from the LP on the homoclinic connections. This curve is given in Figure 11. The elapsed time to compute 53 points was about 42 seconds.

---

**Figure 10.** Fold points on the branch of homoclinic connections of (56)

**Figure 11.** Branch of homoclinic tangencies of (56)
7 Discussion

We have presented new MATLAB tools to continue heteroclinic and homoclinic orbits to fixed points of iterated maps, as well as to compute their fold bifurcation curves, corresponding to the tangencies of the invariant manifolds. The proposed methods work for general $n$-dimensional maps. For the special, but important in applications, case $n = 2$ we described the first MATLAB implementation of known algorithms to grow one-dimensional stable and unstable manifolds of the fixed points of noninvertible maps.

All algorithms are implemented in \texttt{Cl\_MatContM}, a noninteractive toolbox for the bifurcation analysis of maps in MATLAB. We expect to present an interactive version of this software soon.

Currently, before starting a continuation, the user has to extract manually approximations to the homo-/heteroclinic orbits from the computed stable and unstable manifolds. This procedure has to be automated in the future.

Finally, at least for the one-dimensional unstable manifolds, an algorithm exists that can grow them in $n$-dimensional maps. This may also be implemented in \texttt{Cl\_MatContM}.

References


