Computing two-dimensional global invariant manifolds in slow-fast systems

J. P. England, B. Krauskopf & H. M. Osinga
Bristol Centre for Applied Nonlinear Mathematics
Department of Engineering Mathematics, University of Bristol,
Queen's Building, Bristol BS8 1TR, UK

January 2006

Keywords: stable and unstable manifolds, boundary value problems, slow-fast systems

Abstract

We present the GlobalizeBVP algorithm for the computation of two-dimensional stable and unstable manifolds of a vector field. Specifically, we use the collocation routines of Auto to solve boundary problems that are used during the computation to find the next approximate geodesic level set on the manifold. The resulting implementation is numerically very stable and well suited for systems with multiple time scales. This is illustrated with the test-case example of the well-known Lorenz manifold, and with a slow-fast model of a somatotroph cell.

1 Introduction

Knowing global (un)stable manifolds of saddle points and saddle periodic orbits is important for understanding the behaviour of a given dynamical system. For example, stable manifolds may form the boundaries of basins of attraction if the system exhibits multistability. Furthermore, intersections between stable and unstable manifolds typically lead to chaotic dynamics. Generally, these global manifolds cannot be found analytically, so that techniques are required for their numerical approximation. The computation of global stable and unstable manifolds has recently enjoyed renewed interest, and several new algorithms have been developed; see [Krauskopf *et al.*, 2005] for a recent overview.

In this paper we present the GlobalizeBVP algorithm for the computation of two-dimensional (un)stable manifolds of saddle equilibria and saddle periodic orbits in an n-dimensional vector field. (In principle, our method works for higher-dimensional manifolds, but an implementation would need to deal with serious challenges concerning the data structure for the manifold and its visualization.) The specific motivation for the GlobalizeBVP

algorithm comes from the need to compute (un)stable manifolds in systems with multiple time scales. Such systems arise in many applications, including neuroscience and the modelling of chemical reactions or electrical circuits. In its simplest form, the system has two sets of variables that evolve on different time scales, one changing much faster than the others — one also speaks of a slow-fast system; see, for example, [Jones, 1995]. As a result of this slow-fast nature orbits depend very sensitively on their initial condition, so that a small perturbation in the initial condition may lead to a very large difference over the entire orbit. Hence, manifold computations that use shooting techniques to integrate the vector field will suffer from computational problems for slow-fast systems.

The GlobalizeBVP algorithm is based on the method from [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003]. It also computes approximate geodesic level sets, the collection of which represents the computed manifold, and finding a new point on the next geodesic level set is set up as a one-parameter family of boundary value problems. In [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003] we used single shooting as a boundary value solver, but this leads to numerical problems when investigating sensitive systems. The GlobalizeBVP overcomes this difficulty by using the collocation routines of Auto [Doedel, 1981, Doedel *et al.*, 2000] for solving the boundary value problems. Furthermore, it uses Auto's pseudo-arclength pathfollowing routines to follow the solutions. The incorporation of the well-tested routines of Auto makes the GlobalizeBVP algorithm very stable. In particular, this solves the problem of sensitivity to initial conditions as the size of a continuation step is measured over the whole orbit and not just the initial condition. As a result, GlobalizeBVP is indeed able to compute (un)stable manifolds in slow-fast systems. This particular strength of GlobalizeBVP is illustrated in Section 4 for a slow-fast model of a somatotroph cell, for which the previous algorithm that uses shooting breaks down at a very early stage.

The GlobalizeBVP algorithm is very similar in spirit to the ManBVP algorithm for the computation of one-dimensional (un)stable manifolds of a fixed point of a Poincaré map; see [England *et al.*, 2005]. The ManBVP algorithm also uses the collocation and pathfollowing routines of Auto to continue the orbit segments that define the Poincaré map. The begin point is restricted to the part of the one-dimensional manifold that has already been computed. Hence, the other end point, which is restricted to lie in the Poincaré section, traces out a new part of the manifold. The difference with GlobalizeBVP lies in the set-up of the family of boundary value problems that is continued; see Section 2.1.

A different method that also uses continuation of a boundary value problem in Auto for the computation of two-dimensional (un)stable manifolds of vector fields is that by Doedel; see [Krauskopf *et al.*, 2005, Section 3]. His approach involves parameterisation of the manifold by orbits rather than by geodesic level sets. It follows an orbit segment on the manifold of, for example, a particular arclength, while one boundary point is allowed to vary along a circle (or an ellipse) in the (un)stable eigenspace. Doedel's method is also very powerful in the context of slow-fast systems; see [Krogh-Madsen *et al.*, 2004] for an application. However, producing a nicely distributed mesh on the manifold would require post-processing.

This paper is organised as follows. In Section 2 we introduce the GlobalizeBVP algorithm, where we focus in detail on the set-up and the linking with the Auto routines.

Its performance is demonstrated in Section 3 with the Lorenz manifold — the stable manifold of the origin of the well-known Lorenz system. We then show in Section 4 with the example of the model of a somatotroph cell that the GlobalizeBVP algorithm is indeed able to compute (un)stable manifolds in a slow-fast system. Finally, we draw conclusions in Section 5.

2 The GlobalizeBVP algorithm

In its abstract form, the GlobalizeBVP algorithm is able to compute a k-dimensional stable or unstable manifold of a saddle equilibrium or periodic orbit of an n-dimensional vector field. The current implementation computes two-dimensional manifolds in a phase space of any dimension. The main difference between the GlobalizeBVP algorithm and the method in [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003] is how the arising boundary value problems are solved. This difference is key to the GlobalizeBVP algorithm's superior performance in the context of slow-fast systems.

To keep this exposition simple, we concentrate here on computing a two-dimensional unstable manifold $W^u(\mathbf{x}_0)$ of a saddle equilibrium $\mathbf{x}_0$ of the three-dimensional vector field

$$\dot{\mathbf{x}} = f(\mathbf{x}), \tag{1}$$

where $f : \mathbb{R}^3 \to \mathbb{R}^3$ is at least C^2 and $f(\mathbf{x}_0) = 0$. Hence, we assume that $\mathbf{x}_0$ has two unstable eigenvalues $\lambda^u_{1,2}$ with corresponding eigenvectors that span the unstable eigenspace $E^u(\mathbf{x}_0)$. The unstable manifold of $\mathbf{x}_0$ is defined as

$$W^u(\mathbf{x}_0) = \left\{\mathbf{x} \in \mathbb{R}^n \mid \varphi^t(\mathbf{x}) \to \mathbf{x}_0 \text{ as } t \to -\infty\right\},$$

where φ^t is the flow of (1). Similarly, the stable manifold of $\mathbf{x}_0$ is defined as

$$W^s(\mathbf{x}_0) = \left\{\mathbf{x} \in \mathbb{R}^3 \mid \varphi^t(\mathbf{x}) \to \mathbf{x}_0 \text{ as } t \to \infty\right\}.$$

According to the Stable and Unstable Manifold Theorem [Palis and de Melo, 1982], $W^u(\mathbf{x}_0)$ and $W^s(\mathbf{x}_0)$ are smooth manifolds that are tangent at $\mathbf{x}_0$ to $E^u(\mathbf{x}_0)$ and $E^s(\mathbf{x}_0)$, respectively. Furthermore, they are invariant under the flow φ^t. Throughout this paper we use the convention that $W^u(\mathbf{x}_0)$ is red and $W^s(\mathbf{x}_0)$ is blue. When the direction of the flow is reversed by putting a minus sign in front of $f(\mathbf{x})$ the stable manifold and unstable manifolds interchange their roles. Therefore, it suffices to present the algorithm for the computation of $W^u(\mathbf{x}_0)$ only.

We start the computation from a disc in $E^u(\mathbf{x}_0)$ centred at $\mathbf{x}_0$ with a small radius δ. A mesh of N equally-spaced points is chosen on the boundary of this disc, which forms a closed curve C_δ that is an approximation of the geodesic level set on $W^u(\mathbf{x}_0)$ at distance δ from $\mathbf{x}_0$. The algorithm now grows the manifold by adding new approximations of geodesic level sets at larger distances from $\mathbf{x}_0$.

Let the last computed geodesic level set be denoted by C_r and suppose that we wish to find the next geodesic level set C_b, at a distance Δ from C_r. For each point $\mathbf{r}_k$ on C_r we wish to find

	N	δ	$\alpha_{\min}$	$\alpha_{\max}$	$(\Delta\alpha)_{\min}$	$(\Delta\alpha)_{\max}$	$\Delta_{\min}$	Δ_h
Lorenz system	20	1.0	0.3	0.4	0.1	1.0	0.4	2.0
Somatotroph model	20	5.0	0.3	0.4	0.05	0.5	0.01	2.0

Table 1: Accuracy parameters as used in the examples of the Lorenz system in Section 3 and the somatotroph model in Section 4. Here N is the number of meshpoints on the first geodesic level set, which is a circle about $\mathbf{x}_0$ of radius δ; the angle α and the product $\Delta\alpha$ are constrained to lie in $[\alpha_{\min}, \alpha_{\max}]$ and $[(\Delta\alpha)_{\min}, (\Delta\alpha)_{\max}]$, respectively. We always accept a level set at distance $\Delta_{\min}$ and Δ_h is the maximal distance between neighbouring mesh points on the same geodesic level set; see [Krauskopf and Osinga, 1999] for details.

the point $\mathbf{b}_k$ on C_b that lies closest to $\mathbf{r}_k$. To this end, we define a plane $\mathcal{F}_{\mathbf{r}_k}$ (approximately) perpendicular to C_r at $\mathbf{r}_k$. The (unknown) intersection $W^u(\mathbf{x}_0) \cap \mathcal{F}_{\mathbf{r}_k}$ is a one-dimensional curve that is (locally) well defined. Note that any point in $W^u(\mathbf{x}_0) \cap \mathcal{F}_{\mathbf{r}_k}$ lies on an orbit that passes through C_r since, by definition, orbits on $W^u(\mathbf{x}_0)$ come from $\mathbf{x}_0$; see already Figure 1. We find $\mathbf{b}_k \in \mathcal{F}_{\mathbf{r}_k}$ at distance Δ from $\mathbf{r}_k$ as the boundary point of such an orbit segment that has its other boundary point on (the piecewise linear approximation of) C_r. The new point found on the next geodesic level set $\mathbf{b}_k$ is then tested against the standard accuracy criteria as explained in [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003]. Namely, we restrict the maximal allowed angle α between points on three successive geodesic level sets and the product $\Delta\alpha$. This maintains a good resolution of the manifold; if $\mathbf{b}_k$ is not acceptable then the geodesic level set currently being computed is discarded and Δ is reduced. If all points on C_b are found and acceptable then the geodesic level set is added to the manifold.

Geodesic level sets are added until a prescribed geodesic distance is reached. As more geodesic level sets are added, the mesh may need to be adapted such that points are added/removed as required to maintain a good resolution of the surface. We refer to [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003] for details but emphasize that interpolation is always performed only on the geodesic level set C_r, where mesh points are never more than a prescribed distance Δ_h apart. This ensures that the overall accuracy is controlled. The user-specified accuracy parameters for the examples in this paper are shown in Table 1.

2.1 Family of boundary value problems

The crucial step in the GlobalizeBVP algorithm is the solution of the boundary value problem. In [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003] orbit segments are found by single shooting and then bisection is used to obtain a point at distance Δ. Here, we solve boundary value problems by collocation and in combination with pseudo-arclength continuation.

Specifically, we seek orbit segments $\{\mathbf{x}(t), 0 \leq t \leq T\}$ for some $T \geq 0$, that satisfy both the vector field such that $\mathbf{x}(0)$ lies on the computed approximation of the previous geodesic level set C_r, and $\mathbf{x}(T) \in \mathcal{F}_{\mathbf{r}_k}$. There exists a whole family of solutions, parameterised by

T, starting with $T = 0$ and the solution $\mathbf{x}(t) \equiv \mathbf{r}_k$. Our goal is to continue this starting solution for $T > 0$ where T is such that the associated solution has the additional property $||\mathbf{x}(T) - \mathbf{r}_k|| = \Delta$. To this end we rescale the vector field (as is common in AUTO) to

$$\mathbf{u}'(t) = T f(\mathbf{u}(t)), \tag{2}$$

where $\mathbf{u}(t) = \mathbf{x}(tT)$, $0 \leq t \leq 1$ is the rescaled solution. Now the boundary value problem is always restricted to the interval [0,1] so that we have the boundary conditions

$$\mathbf{u}(0) \in C_r, \tag{3}$$
$$\mathbf{u}(1) \in \mathcal{F}_{\mathbf{r}_k}. \tag{4}$$

The point $\mathbf{b}_k$ is determined by continuation in the parameter T, where we monitor the distance

$$\Delta_T = \|\mathbf{u}(1) - \mathbf{r}_k\|$$

between the end boundary point $\mathbf{u}(1)$ and the meshpoint $\mathbf{r}_k$. The continuation starts from the trivial solution $\mathbf{u}(t) = \mathbf{r}_k$, $0 \leq t \leq 1$ with $T = 0$. As T increases, we monitor Δ_T until the required distance Δ is reached, which is detected as a zero of a user-defined function in AUTO; see already Section 2.4.

During the continuation, as the integration time T increases, $\mathbf{u}(0)$ varies along the linear segment between two mesh points on the geodesic level set C_r, and $\mathbf{u}(1)$ traces out the local intersection of $W^u(\mathbf{x}_0)$ with $\mathcal{F}_{\mathbf{r}_k}$ until it lies at distance Δ from the base point $\mathbf{r}_k$. Note that locally near $\mathbf{r}_k$, $\mathbf{u}(0)$ only varies along segments that lie on one side of $\mathcal{F}_{\mathbf{r}_k}$. We decide at which side to start the continuation by assessing the direction of the vector field at $\mathbf{r}_k$. Let us assume that, initially, $\mathbf{u}(0)$ traces along the segment between $\mathbf{r}_{k-1}$ an $\mathbf{r}_k$.

Figure 1 illustrates several cases for the local behaviour of the family of solutions of (2)–(4). Panel (a) shows the simplest typical case: as a continuation step is made, the integration time T increases from $T = 0$ and $\mathbf{u}(0)$ moves along the line segment between $\mathbf{r}_{k-1}$ and $\mathbf{r}_k$. The endpoint $\mathbf{u}(1)$ also moves away from $\mathbf{r}_k$ in $\mathcal{F}_{\mathbf{r}_k}$ tracing out the intersection of $W^u(\mathbf{x}_0)$ with $\mathcal{F}_{\mathbf{r}_k}$. The continuation stops as soon as $\mathbf{u}(1)$ lies at Δ from $\mathbf{r}_k$. In Figure 1 this happens before $\mathbf{u}(0)$ reaches the end of the line segment $(\mathbf{r}_{k-1}, \mathbf{r}_k)$. If $\mathbf{u}(0) = \mathbf{r}_{k-1}$ and still $\|\mathbf{u}(1) - \mathbf{r}_k\| < \Delta$, the boundary condition (3) is reset such that $\mathbf{u}(0)$ now lies on the next segment $(\mathbf{r}_{k-2}, \mathbf{r}_{k-1})$, and so on. Panels (b) and (c) show more complicated cases where tangencies with C_r or $\mathcal{F}_{\mathbf{r}_k}$ occur. These are explained in more detail in the next section.

2.2 Tangencies of orbit segments

The continuation of the boundary value problem in AUTO is not only very accurate, but also deals effectively with two potential problems illustrated in Figures 1(b) and (c). Figure 1(b) shows the case when the flow becomes tangent to $\mathcal{F}_{\mathbf{r}_k}$. Up until the tangency, the situation would be as in Figure 1(a). As $\mathbf{u}(1)$ moves through the tangency the orbit gains an extra intersection with $\mathcal{F}_{\mathbf{r}_k}$. During the pseudo-arclength continuation of AUTO this happens automatically while $\mathbf{u}(0)$ moves backward towards $\mathbf{r}_k$. In fact, it is possible that $\mathbf{u}(0)$ moves

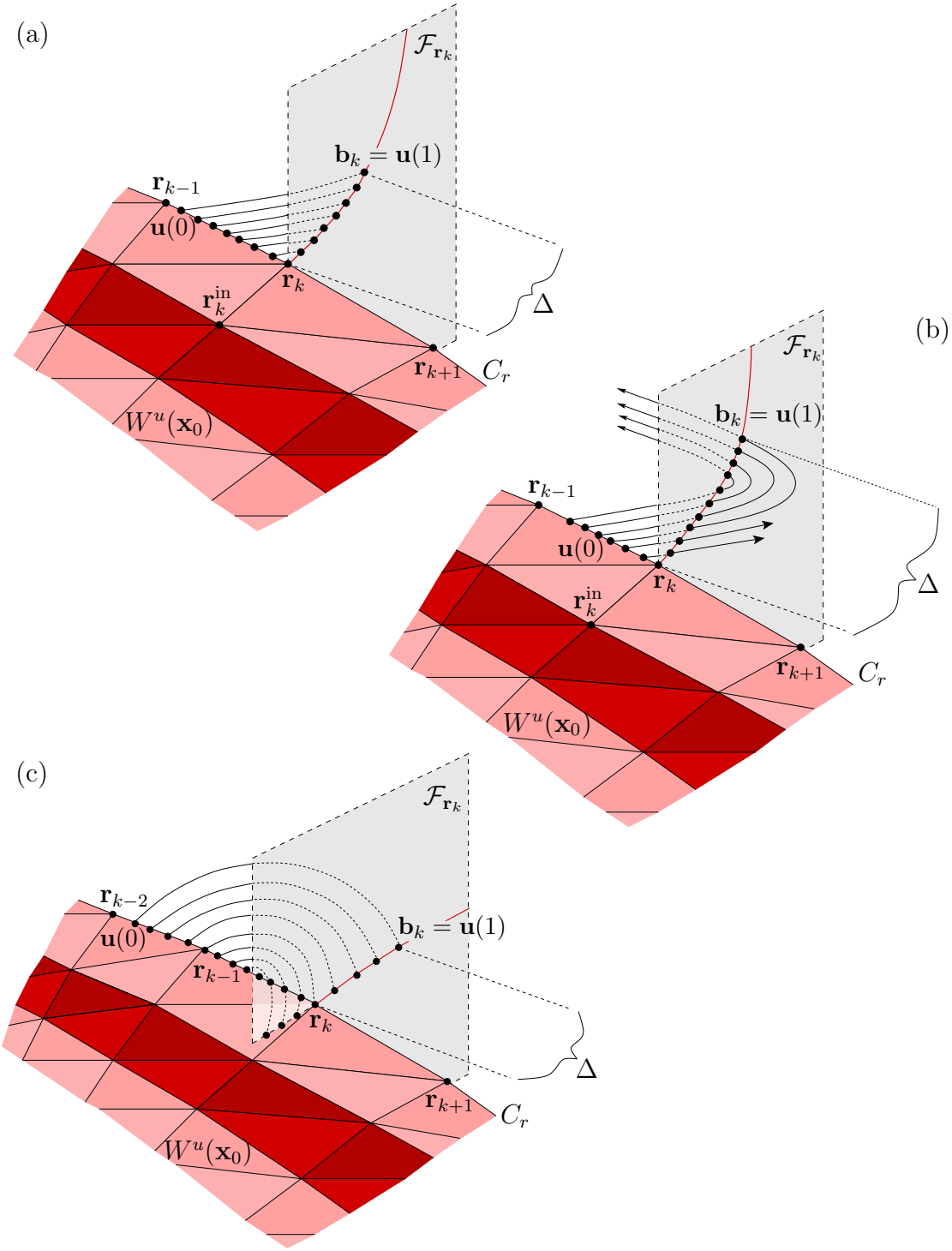

Figure 1: An illustration of finding a new point at distance Δ from a point $\mathbf{r}_k$ on the geodesic level set C_r in the GlobalizeBVP algorithm. The boundary value problem is formulated such that $\mathbf{u}(0)$ lies on the line segment between $\mathbf{r}_k$ and $\mathbf{r}_{k-1}$ and $\mathbf{u}(1)$ lies in $\mathcal{F}_{\mathbf{r}_k}$. Panel (a) is the simplest case, panel (b) shows the case when the flow becomes tangent to $\mathcal{F}_{\mathbf{r}_k}$, and panel (c) shows the case when the flow becomes tangent to C_r; note that in panel (c) $\mathbf{u}(0)$ passes through $\mathbf{r}_{k-1}$, after which boundary condition (3) is reset to the line segment between $\mathbf{r}_{k-1}$ and $\mathbf{r}_{k-2}$.

all the way back to $\mathbf{r}_k$, in which case the continuation switches to the line segment $(\mathbf{r}_k, \mathbf{r}_{k+1})$ on the other side of $\mathcal{F}_{\mathbf{r}_k}$.

In Figure 1(c) another possible situation is illustrated, namely that the flow becomes tangent to C_r. Now, the endpoint $\mathbf{u}(1) \in \mathcal{F}_{\mathbf{r}_k}$ of the orbit segment initially moves back along the manifold that has already been computed. As $\mathbf{u}(0)$ passes the tangency point on C_r, the endpoint $\mathbf{u}(1)$ changes direction and the computation continues until $\mathbf{u}(1)$ is at distance Δ from $\mathbf{r}_k$ in the correct direction. Hence, it is important to include a sign when measuring $\|\mathbf{u}(1) - \mathbf{r}_k\|$, because $\mathbf{u}(1)$ may be at distance Δ from $\mathbf{r}_k$ but in the wrong direction on a band of the manifold already computed; see also Section 2.4.

The method in [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003] already deals with the two tangency situations above, but not in an optimal way. In particular, a tangency has to be detected and the initial value solver adjusted to account for the additional intersections with $\mathcal{F}_{\mathbf{r}_k}$. The GlobalizeBVP implementation does not require any detection of such tangencies, as the orbit segments move through them continuously. The case shown in Figure 1(c), where the direction in which $\|\mathbf{u}(1) - \mathbf{r}_k\|$ is measured changes as one moves through the tangency, typically occurs when $\mathbf{x}_0$ has a complex conjugate pair of eigenvalues. Due to the spiralling of the orbits, $\mathbf{u}(0)$ may trace out such a long part of C_r, that a segment other than $(\mathbf{r}_{k-1}, \mathbf{r}_k)$ and $(\mathbf{r}_k, \mathbf{r}_{k+1})$ is reached which also intersects $\mathcal{F}_{\mathbf{r}_k}$. This causes serious problems with the shooting method used in [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003], while the continuation in T of the GlobalizeBVP implementation is not affected at all. Note that we do not require the flow to be directed outward as is required for the method of [Guckenheimer and Worfolk, 1993].

2.3 Boundary Conditions

In order to restrict the begin point $\mathbf{u}(0)$ to a linear segment on C_r in n-dimensional space, and monitor its position along this segment, we need n boundary conditions.

Let us assume that we wish to restrict $\mathbf{u}(0)$ to the line segment,

$$L(\tau) = \tau \mathbf{r}_{i-1} + (1 - \tau)\mathbf{r}_i, \quad 0 \leq \tau \leq 1.$$

It seems straightforward to consider the n Cartesian components of $L(\tau) = [L_1(\tau), \ldots, L_n(\tau)]$ and require the boundary conditions

$$u_j(0) - L_j(\tau) = 0, \quad 1 \leq j \leq n,$$

where $\mathbf{u}(0) = [u_1(0), \ldots, u_n(0)]$. The position of $\mathbf{u}(0)$ is then given by the value of τ. However, with every step in τ, all n boundary conditions change and if the solution is corrected to satisfy one, then the others will typically also change.

Therefore, our implementation uses $n - 1$ boundary conditions to restrict $\mathbf{u}(0)$ to the segment $L(\tau)$ and one to fix its position on $L(\tau)$ as a function of the parameter τ. This is illustrated in Figure 2 for $n = 3$, where two orthogonal planes, Σ_2 and Σ_3, intersect along the line segment $L(\tau)$ between the two mesh points $\mathbf{r}_{i-1}$ and $\mathbf{r}_i$ on C_r. In general, the first $n-1$ boundary conditions restrict $\mathbf{u}(0)$ to $(n-1)$-dimensional hyperplanes that are defined using

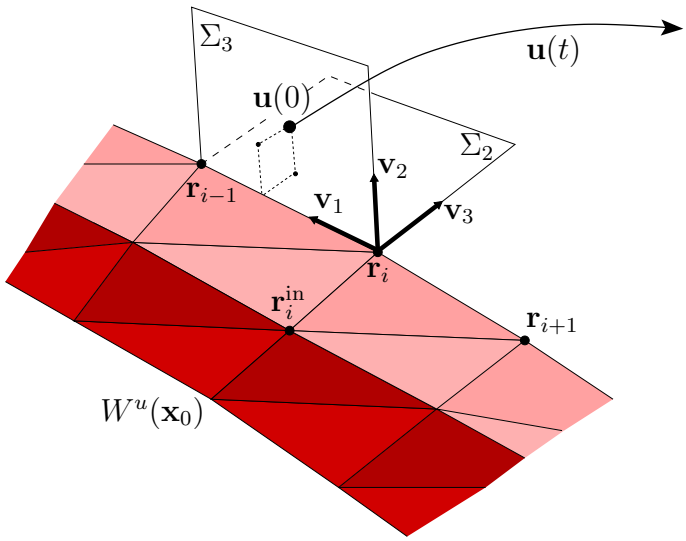

Figure 2: An illustration of the boundary condition set-up to restrict $\mathbf{u}(0)$ to the line segment $L(\tau)$ for $n = 3$. Two planes, Σ_2 and Σ_3, intersect orthogonally along $L(\tau)$. The boundary conditions minimise the distance of $\mathbf{u}(0)$ from both Σ_2 and Σ_3 independent of the parameter τ.

an orthogonal basis of normal vectors that span the orthogonal complement of the direction $L(\tau)$. Then these boundary conditions only depend on $\mathbf{r}_i$ and $\mathbf{r}_{i-1}$, but not on τ.

The case for $n = 3$ is straightforward. We set up a coordinate system of three orthonormal vectors $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, where $\mathbf{v}_1$ is defined as the unit vector from $\mathbf{r}_i$ to $\mathbf{r}_{i-1}$. The vectors $\mathbf{v}_2$ and $\mathbf{v}_3$ that span the orthogonal complement are found by computing appropriate cross products. For $n > 3$, the orthonormal basis defining the hyperplanes Σ_i, $i = 2, \ldots, n$ can be determined, for example, by rotating the standard coordinate axes $(\mathbf{e}_1, \ldots, \mathbf{e}_n)$ such that $\mathbf{e}_1$ aligns with $\mathbf{v}_1$. Note that there is some 'freedom' in how the basis is chosen, but the choice is not important in the execution of the algorithm.

We restrict $\mathbf{u}(0)$ to the line segment $L(\tau)$ by enforcing the boundary conditions

$$\langle (\mathbf{u}(0) - \mathbf{r}_i), \mathbf{v}_i \rangle = 0, \quad 2 \leq j \leq n,$$

where $\langle \cdot, \cdot \rangle$ denotes the dot product.

The nth boundary condition restricts $\mathbf{u}(0)$ to a point on $L(\tau)$. We can define the actual distance of a point along $L(\tau)$ as

$$\tau_T = \frac{\langle \mathbf{u}(0) - \mathbf{r}_i, \mathbf{v}_1 \rangle}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|},$$

which gives rise to the third boundary condition

$$\tau_T - \tau = 0.$$

NDIM, dimension		n
NBC, boundary conditions:		$n+1$
restriction $\mathbf{u}(0)$ to $L(\tau)$	$n-1$	
distance $\mathbf{u}(0)$ on $L(\tau)$	1	
restriction $\mathbf{u}(1)$ to $\mathcal{F}_{\mathbf{r}_k}$	1	
NICP, free parameters: T and τ		2
NBC $-$ NDIM $+ 1 -$ NICP $=$		0

Table 2: The number of boundary conditions and free parameters required by AUTO for a well-posed boundary value problem.

The parameter τ is a free parameter, which means that in each continuation step the orbit is corrected such that $\tau_T = \tau$.

Only one boundary condition is required to restrict $\mathbf{u}(1)$ to the plane $\mathcal{F}_{\mathbf{r}_k}$. Namely,

$$\langle \mathbf{u}(1) - \mathbf{r}_k + \mathbf{v}_F, \hat{n} \rangle = 0,$$

where,

$$\hat{n} = \frac{\mathbf{r}_{k+1} - \mathbf{r}_{k-1}}{\|r_{k+1} - \mathbf{r}_{k-1}\|}$$

is the normal of the plane $\mathcal{F}_{\mathbf{r}_k}$ and $\mathbf{v}_F$ is an arbitrary vector in $\mathcal{F}_{\mathbf{r}_k}$. We use $\mathbf{v}_F = (\mathbf{r}_k - \mathbf{r}_k^{\text{in}}) \times \hat{n}$ for $n = 3$. The vector $\mathbf{v}_F$ is needed to ensure that the dot product is zero only when $\mathbf{u}(1) \in \mathcal{F}_{\mathbf{r}_k}$. Note that, initially, we start with $\mathbf{u}(t) \equiv \mathbf{r}_k$ and thus $\mathbf{u}(1) - \mathbf{r}_k = 0$. Without $\mathbf{v}_F$ the boundary condition would then be satisfied automatically; AUTO cannot start from such a degenerate solution.

The total number of boundary conditions and parameters for this continuation problem is summarised in Table 2. The boundary value problem is well posed if the number of free parameters (NICP) equals one plus the difference between the number of boundary conditions (BCN) and the dimension (NDIM). That is, NICP = NBC $-$ NDIM $+ 1$. In our implementation, the free parameters are always the period T and the distance τ along the respective line segment of C_r.

2.4 User-defined functions

We define user-defined (UZ) functions to monitor specific quantities, so that we can switch to a new segment as soon as $\mathbf{u}(0)$ reaches $\mathbf{r}_i$ or $\mathbf{r}_{i+1}$. Furthermore, we wish to monitor the distance Δ between $\mathbf{u}(1)$ and $\mathbf{r}_k$. As mentioned in Section 2.1 we must track which side of C_r the end point $\mathbf{u}(1)$ is on. Hence, we cannot simply monitor

$$\|\mathbf{u}(1) - \mathbf{r}_k\|.$$

Instead, we define

$$\text{UZ}(0) = \|\mathbf{u}(1) - \mathbf{r}_k\| \, \frac{\langle \mathbf{v}_r, \mathbf{u}(1) - \mathbf{r}_k \rangle}{|\langle \mathbf{v}_r, \mathbf{u}(1) - \mathbf{r}_k \rangle|}, \tag{5}$$

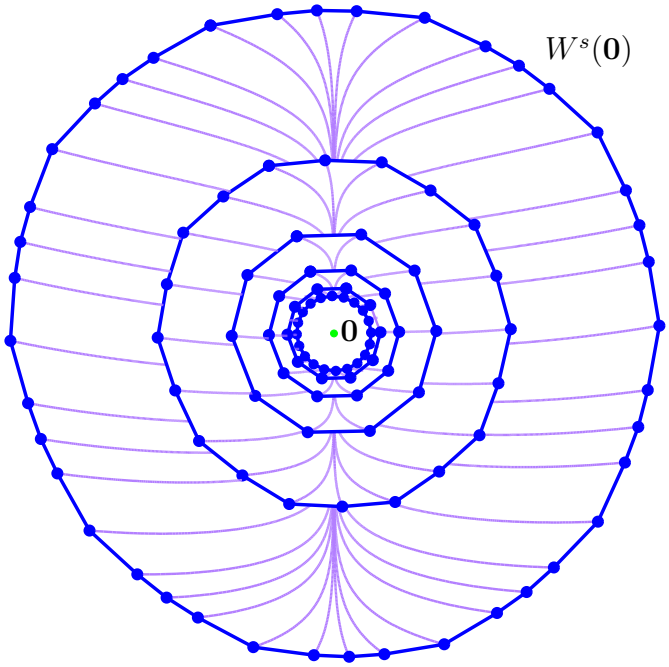

Figure 3: Data of the computation of five geodesic level sets of the Lorenz manifold up to geodesic distance 8.75. The computed geodesic level sets of the manifold are shown in blue and each of the point on the geodesic level set is an endpoint of a computed orbit segment (shown in light purple).

where $\mathbf{v}_r = \mathbf{r}_k - \mathbf{r}_k^{\text{in}}$, is a vector that points away from the part of $W^u(\mathbf{x}_0)$ computed so far. Note that (5) is not defined at the start of the continuation, but this does not affect the computation; as soon as one continuation step is made, (5) has a value. We are only interested when $\text{UZ}(0) = \Delta$ and so for computational purposes, it does not matter if UZ(0) is undefined away from Δ.

We also require user-defined functions to monitor the distance τ at which $\mathbf{u}(0)$ lies along $L(\tau)$. If $\tau = 0$ or $\tau = 1$ then we need to stop and reset the continuation along the next/previous segment as required. Hence, we define

$$\text{UZ}(1) = \tau, \tag{6}$$
$$\text{UZ}(2) = \tau - 1. \tag{7}$$

If (5) holds, we check the accuracy condition for the candidate $\mathbf{b}_k = \mathbf{u}(1)$. If either (6) or (7) hold, we switch segments and set $L(\tau) = \tau\mathbf{r}_{i-1} + (1-\tau)\mathbf{r}_i$ or $L(\tau) = \tau\mathbf{r}_{i+1} + (1-\tau)\mathbf{r}_{i+2}$, respectively.

3 The Lorenz manifold

To demonstrate the GlobalizeBVP implementation we compute the stable manifold of the origin in the well-known Lorenz system [Lorenz, 1963], which is defined as

$$\begin{aligned} \dot{x} &= \sigma(y-x), \\ \dot{y} &= -\rho x - y - xz, \\ \dot{z} &= xy - \beta z. \end{aligned} \tag{8}$$

We use the standard values for the parameters: $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. The computation of the two-dimensional stable manifold of this system, called the Lorenz manifold, is widely used as a benchmark with which to compare manifold computation algorithms; see [Krauskopf *et al.*, 2005].

The origin $\mathbf{x}_0$ is a saddle equilibrium with two stable eigenvalues $\lambda_1^s \approx -22.828$ and $\lambda_2^s = \beta \approx -2.667$ and an unstable eigenvalue $\lambda^u \approx 11.828$. The two stable eigenvalues have corresponding eigenvectors that span the stable eigenspace $E^s(\mathbf{x}_0)$. The stable eigenvector associated with λ_2^s lies on the z-axis which is, in fact, invariant under the flow.

Figure 3 shows the orbits (light purple) used in the computation of the first five geodesic level sets (blue) of the Lorenz manifold up to geodesic distance 8.75. For the accuracy parameters used in this calculation see Table 1. All points on the computed manifold are end points of orbit segments whose beginning points lie on a line segment between two points on the previous geodesic level set. Note that the z-axis is the vertical line through the origin in this picture. One can clearly see that many initial conditions start very close to the z-axis due to the strong stretching transverse to this axis; on purpose, the mesh points on C_δ were taken to lie off the z-axis.

Figure 4 shows four views of the orbits used in computing the Lorenz manifold up to geodesic distance 100.75. Panels (a) and (b) show two projections of the entire manifold and panels (c) and (d) show two enlarged views. Notice in panel (d) how the flow is nearly tangent along parts of the outer geodesic level sets.

Figure 5 shows some of the orbit segments used in the continuation to find a particular new point on the next geodesic level set of the Lorenz manifold; compare Figure 1(a) and (c). The plane $\mathcal{F}_{\mathbf{r}_k}$ is shown in green and the computed approximate geodesic level sets are shown by blue lines. As before, the orbits used in the continuation are light purple. Panel (a) shows real data of the case illustrated in Figure 1(a). The beginning boundary point $\mathbf{u}(0)$ is varied from the base point $\mathbf{r}_k$ firstly along the line segment between $\mathbf{r}_{k-1}$ and $\mathbf{r}_k$. When it reaches $\mathbf{r}_{k-1}$ the boundary conditions are changed such that $\mathbf{u}(0)$ then varies along the line segment between $\mathbf{r}_{k-2}$ and $\mathbf{r}_{k-1}$ until $\mathbf{u}(1)$ lies at distance Δ from $\mathbf{r}_k$ in $\mathcal{F}_{\mathbf{r}_k}$. This final, accepted orbit is shown in black and its endpoint is tested for accuracy to be added to the manifold. Notice how the orbit segments are closer together as $\mathbf{u}(0)$ approaches $\mathbf{r}_{k-1}$; this is due to the convergence toward the user-defined function (6) in the continuation.

Panel (b) of Figure 5 shows computed orbit segments for the case illustrated in Figure 1(c). Initially, as $\mathbf{u}(0)$ varies along C_r, the endpoint $\mathbf{u}(1)$ moves in the 'wrong direction' over a part of the manifold that was already computed. As $\mathbf{u}(0)$ moves through a tangency point of the flow with C_r (marked by a black dot), the direction in which $\mathbf{u}(1)$ is moving

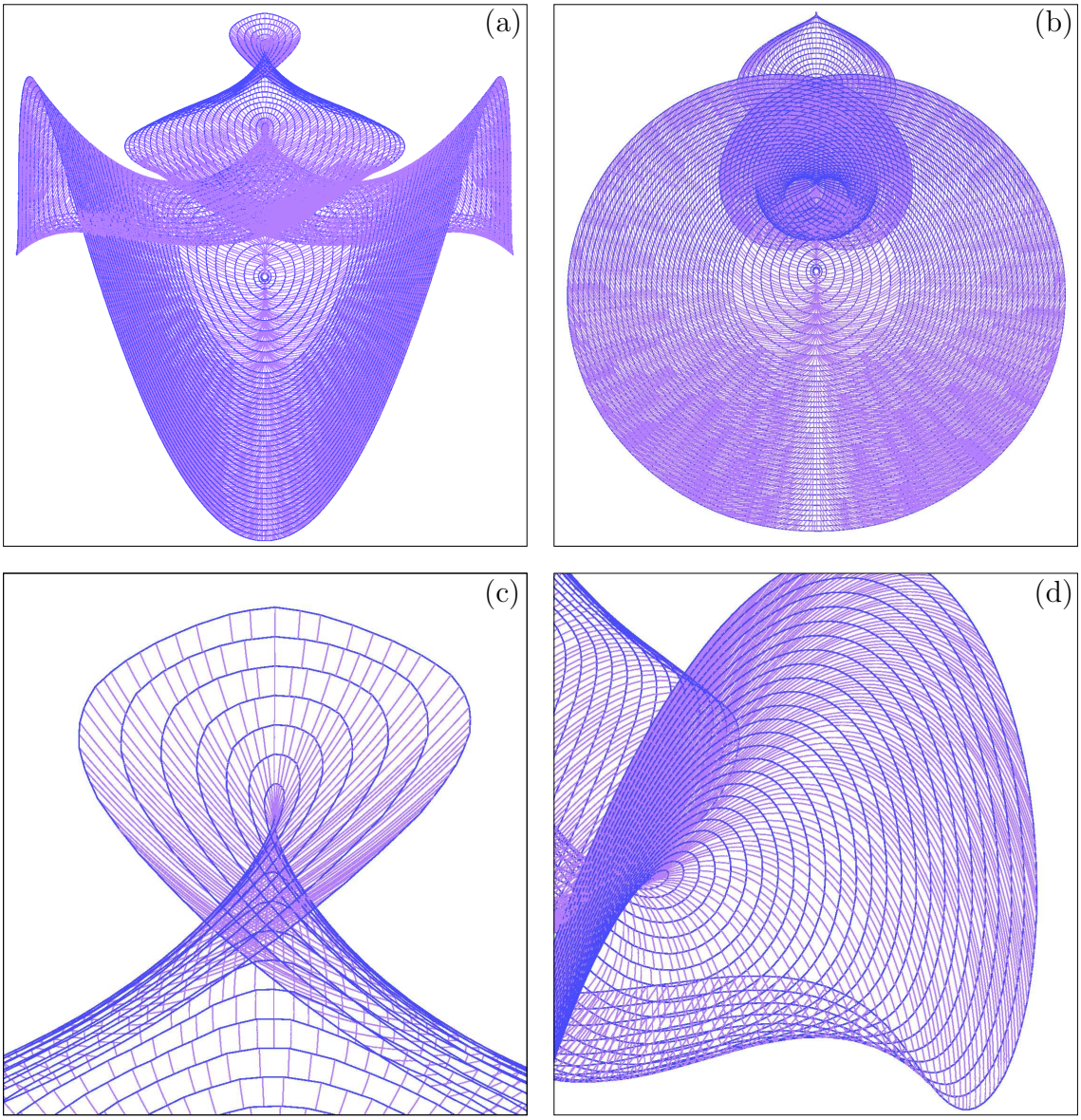

Figure 4: The orbit segments (light purple) used in computing the Lorenz manifold up to geodesic distance 100.75. The approximate geodesic level sets are shown in blue. The z-axis is the vertical axis in the middle of panels (a)–(c).

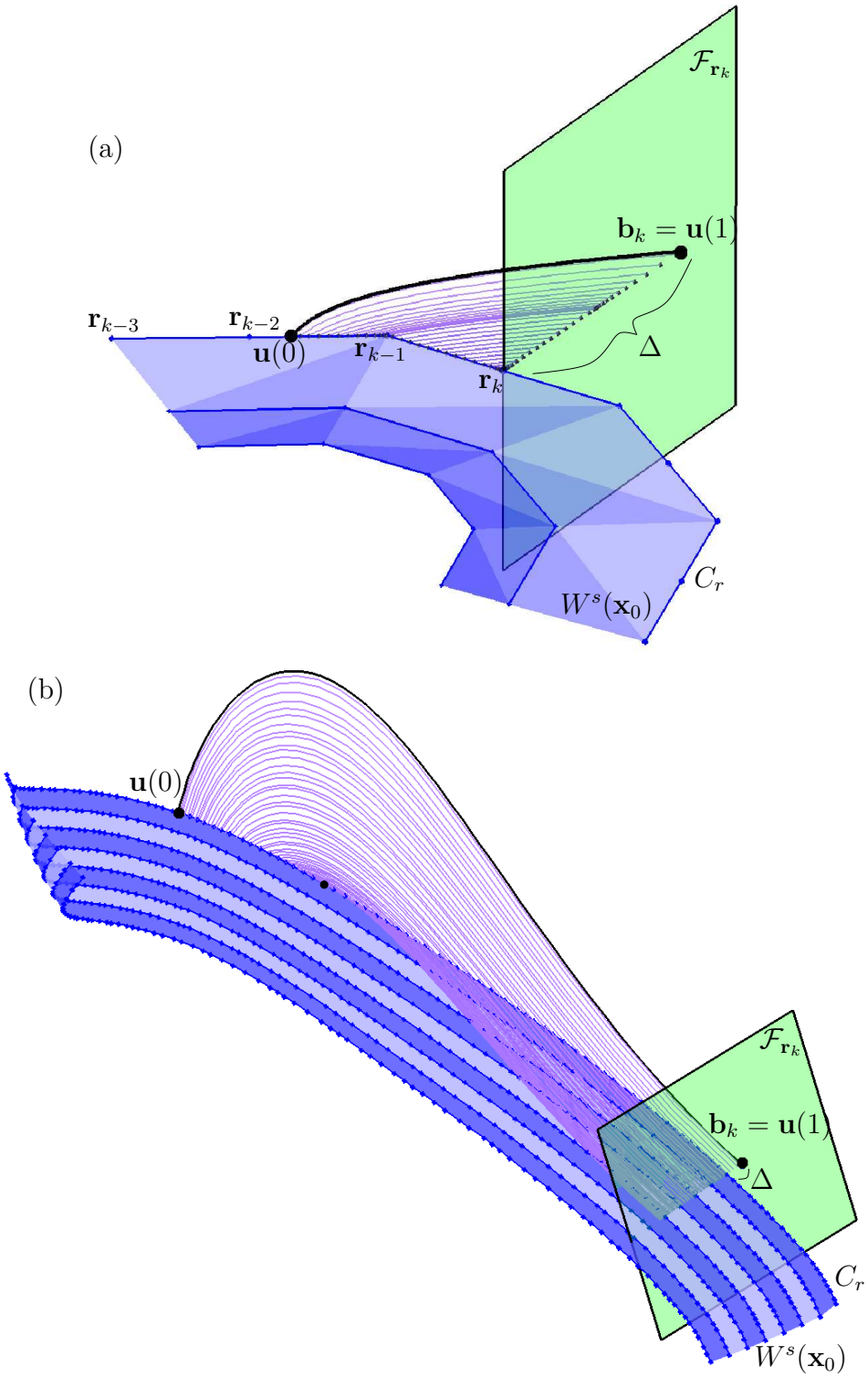

Figure 5: Data of the orbit segments used in the continuation to find a new point on the Lorenz manifold. The black orbit has its end boundary point at distance Δ from the point on the previous geodesic level set. Panel (a) shows the simple situation sketched in Figure 1(a), and panel (b) a tangency with C_r as was sketched in Figure 1(c).

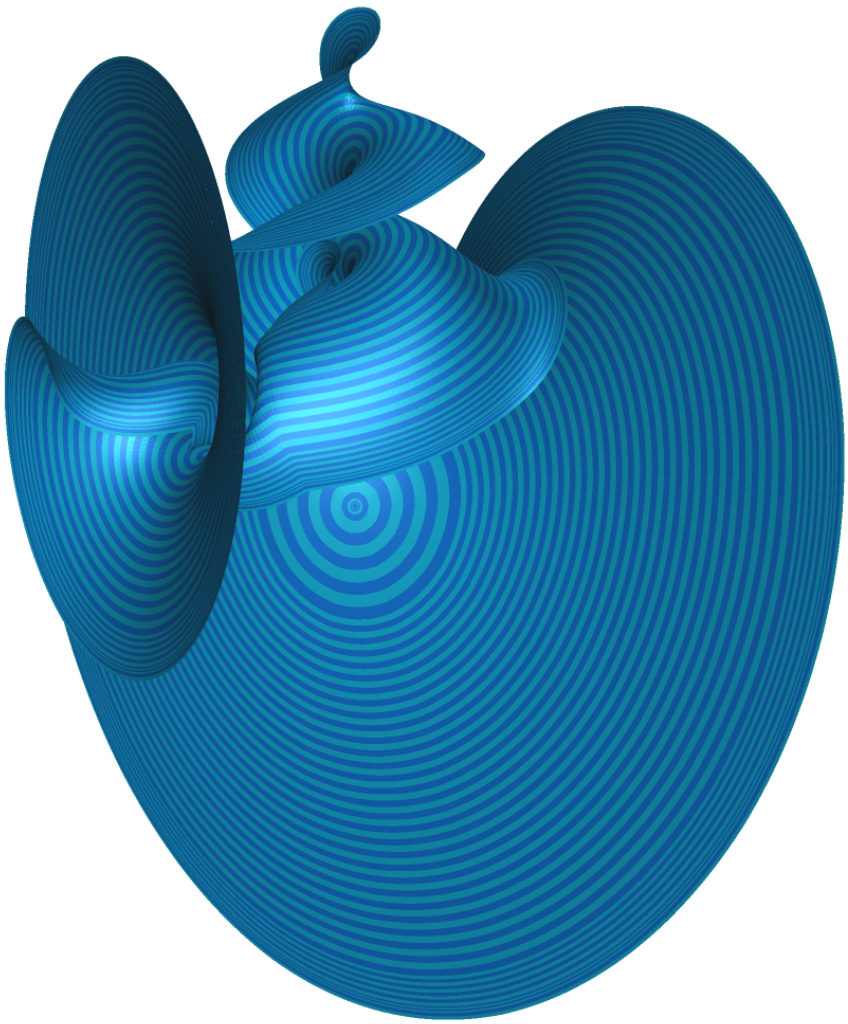

Figure 6: The Lorenz manifold computed up to geodesic distance 161.75 with the GlobalizeBVP implementation shown from the same view as in [Krauskopf *et al.*, 2005].

changes and the continuation progresses until $\mathbf{u}(1)$ is at distance Δ from $\mathbf{r}_k$. Again, the final orbit is shown in black.

Figure 6 shows the Lorenz manifold computed with the GlobalizeBVP algorithm up to geodesic distance 161.75. The colour of the geodesic level sets is alternated to demonstrate how the manifold is grown. Previously, it could only be computed up to a geodesic distance 154.75, when the computation stopped; for example, see [Osinga and Krauskopf, 2002] for illustrations and animations. This appears to happen been due to inaccuracies of the single shooting implementation for solving the boundary value problems. This is now confirmed by the fact that the GlobalizeBVP algorithm can compute the Lorenz manifold further.

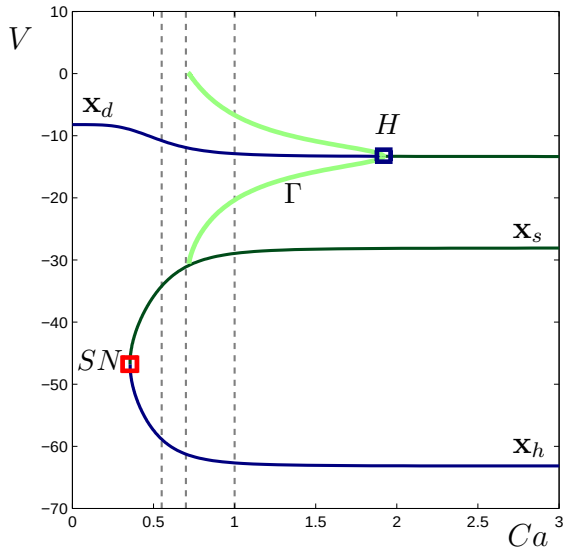

Figure 7: Bifurcation diagram of V versus Ca for the somatotroph model (9). Stable equilibria are shown in blue and saddle equilibria are in dark green. The saddle periodic orbits are light green; both maximal and minimal values of the periodic orbits are drawn. The dashed grey lines indicate the parameter values $Ca = 0.55$, $Ca = 0.71672$ and $Ca = 1.0$ for Figures 9–11.

4 Somatotroph cell model

The somatotroph model is a three-dimensional system describing the voltage potential across the cell membrane of a neuro-endocrine cell depending on the opening and closing dynamics of certain gating variables. The model in so-called fast subsystem form is given as

$$\begin{cases} \dfrac{dV}{dt} &= \dfrac{-1}{c_m}\,[I_{\mathrm{Ca}}(V,m) + I_{\mathrm{K}}(V,n) \\ & \qquad +I_{\mathrm{KCa}}(V,Ca) + I_{\mathrm{L}}(V)], \\ \dfrac{dm}{dt} &= \dfrac{m_\infty(V) - m}{\tau_m}, \\ \dfrac{dn}{dt} &= \dfrac{n_\infty(V) - n}{\tau_n}, \end{cases} \tag{9}$$

Here V is the voltage and m and n are the gating variables representing probabilities that ionic channels are open. The functions I_{Ca}, I_K, I_{KCa}, I_L, m_∞, and n_∞ are complicated expressions involving many parameters that are tuned to experimental data; see [Osinga and England, 2005] for further details. The dynamics is organised by the modulation of the Ca concentration in the cell. This concentration changes relatively slowly compared to the other variables, and in the fast subsystem model Ca is considered a fixed parameter in the system. The fast subsystem still operates on two different time scales. The variation of the voltage V is about 300 times faster than the variation of the gating vari-

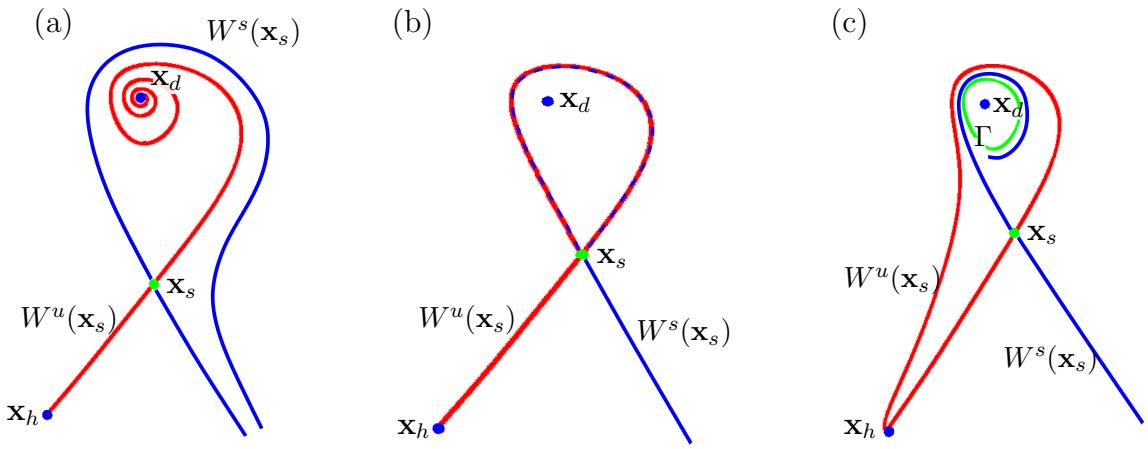

Figure 8: A two-dimensional sketch of the cases before (a), at (b), and after (c) the homoclinic bifurcation. In each panel, the saddle equilibrium is indicated by a green dot and the stable equilibria by blue dots. The unstable manifolds are projections of the real three-dimensional unstable manifolds, in red, and the stable manifolds are sketches, in blue. The periodic orbit Γ is the green closed loop in panel (c).

ables m and n. This causes serious sensitivity in the numerical integration of the systems, which greatly affects the computation of the manifolds with the shooting-based method of [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003].

The bifurcation diagram in Figure 7 shows that for very small Ca-values there is only one equilibrium, the so-called depolarised state $\mathbf{x}_d$ and it is stable. At $Ca \approx 0.335$ a saddle-node bifurcation (denoted SN) occurs that gives birth to a saddle equilibrium $\mathbf{x}_s$ and another stable equilibrium, the hyperpolarised state $\mathbf{x}_h$. At $Ca \approx 1.919$ a subcritical Hopf bifurcation (denoted H) occurs that gives rise to periodic orbits of saddle-type. As Ca is decreased from the Hopf bifurcation, the saddle periodic orbit Γ grows until it is destroyed in a homoclinic bifurcation at $Ca \approx 0.71672$. At this value, the one-dimensional unstable manifold $W^u(\mathbf{x}_s)$ of $\mathbf{x}_s$ is entirely contained in its two-dimensional stable manifold $W^s(\mathbf{x}_s)$. This homoclinic bifurcation is well known in planar systems of this type; see, for example, [Sherman, 1997], but note that, here, the Hopf bifurcation is subcritical instead of supercritical.

System (9) is investigated in [Osinga and England, 2005] by means of computing one-dimensional manifolds of saddle periodic orbits in a suitable Poincaré section with the MANBVP algorithm of [England *et al.*, 2005]. Here we compute relevant two-dimensional stable manifolds with the GLOBALIZEBVP algorithm. Due to the large difference in ranges between the variables (V varies at least between -70 and -10mV, while m and n are by definition contained in $[0, 1]$), we use a weighted norm in the computations such that all the variables are of order 1. More precisely, m is multiplied by a factor of 100, and n is multiplied by 500. This is important when we consider the geodesic distance in the computations.

Let us first compare the situation as sketched in Figure 8 in a two-dimensional representation with the situation given in [Sherman, 1997, Figure 10.5]. The unstable manifolds (red)

in Figure 8 are projections of the real data, while the stable manifolds (blue) are sketches for the two-dimensional case. Panel (a) shows the case for $Ca < 0.71672$, where one branch of the unstable manifold spirals into the stable equilibrium $\mathbf{x}_d$ and the other branch goes to $\mathbf{x}_h$. The stable manifold $W^s(\mathbf{x}_s)$ separates the basins of attraction of $\mathbf{x}_d$ and $\mathbf{x}_h$. Panel (b) shows the situation at the homoclinic bifurcation: the basin boundary of $\mathbf{x}_d$ is now given by only one of the two branches of $W^s(\mathbf{x}_s)$, namely the branch that is homoclinic to $\mathbf{x}_s$. Note that the branch of $W^u(\mathbf{x}_s)$ that converged to $\mathbf{x}_d$ now goes to $\mathbf{x}_s$. This branch coincides with one branch of $W^s(\mathbf{x}_s)$ such that points on it tend toward $\mathbf{x}_s$ in both forward and backward time. Panel (c) shows the case for $Ca > 0.71672$; both branches of $W^u(\mathbf{x}_s)$ now converge to $\mathbf{x}_h$ and $W^s(\mathbf{x}_s)$ no longer separates the two basins of attraction, but spirals in around the saddle periodic orbit Γ. Only points inside Γ tend toward $\mathbf{x}_d$.

In three dimensions the picture is more complicated. In Figures 9–11 we consider the three situations before, approximately at, and after the homoclinic bifurcation, respectively. Specifically we choose $Ca = 0.55$, $Ca = 0.71672$ and $Ca = 1.0$ as is indicated by the dashed grey lines in Figure 7. In Figures 9–11 we show the stable and unstable manifolds $W^s(\mathbf{x}_s)$ and $W^u(\mathbf{x}_s)$ of the saddle equilibrium $\mathbf{x}_s$ and the stable manifold $W^s(\Gamma)$ of the saddle periodic orbit Γ (when it exists, that is, for $Ca = 1.0$). Each figure shows the same two views in the right and left columns. In the top row the two-dimensional manifold $W^s(\mathbf{x}_s)$ is a shaded rendering, while in the bottom row it is rendered semi-transparent. The coloring of $W^s(\mathbf{x}_s)$ changes from dark blue to light blue to help convey its three-dimensional geometry. For the accuracy parameters used in these calculations see Table 1.

Figure 9 shows the situation before the homoclinic bifurcation, namely for $Ca \approx 0.55$. The two-dimensional stable manifold $W^s(\mathbf{x}_s)$ is the separatrix between the basins of attraction of $\mathbf{x}_d \approx (-10.779, 76.585, 61.064)$ and $\mathbf{x}_h \approx (-58.853, 5.619, 0.171)$. The saddle equilibrium $\mathbf{x}_s$ lies approximately at $(-34.225, 31.674, 3.684)$ with eigenvalues $\lambda^s_1 \approx -402.905$, $\lambda^s_2 \approx -45.273$ and $\lambda^u \approx 75.450$. Figure 9 shows $\mathbf{x}_d$, $\mathbf{x}_s$, $\mathbf{x}_h$, the one-dimensional unstable manifold $W^u(\mathbf{x}_s)$ in red, and the two-dimensional stable manifold $W^s(\mathbf{x}_s)$ in blue — computed up to geodesic distance 269.5 with GlobalizeBVP. We remark that the shooting-based method in [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003] was only able to compute this manifold up to geodesic distance 57. In fact, our computation did not break down but was simply stopped at geodesic distance 269.5 once a good impression of the geometry of $W^s(\mathbf{x}_s)$ was obtained. From the applications point of view, the situation in Figure 9 is as follows. One side of $W^u(\mathbf{x}_s)$ goes to the hyperpolarised state $\mathbf{x}_h$ (lower blue dot) and the other spirals into the depolarised state $\mathbf{x}_d$ (upper blue dot). The space inside the part of $W^s(\mathbf{x}_s)$ that is 'folded over' is the basin of attraction of $\mathbf{x}_d$; compare with Figure 8(a).

Figure 10 shows the situation at the homoclinic bifurcation, for $Ca \approx 0.71672$, when there is a unique orbit that tends to $\mathbf{x}_s$ in both forward and backward time, which is in fact one branch of $W^u(\mathbf{x}_s)$. Here, $\mathbf{x}_d = (-11.996, 74.718, 53.367)$ and $\mathbf{x}_h = (-61.416, 4.589, 0.124)$; the saddle $\mathbf{x}_s = (-30.883, 37.973, 5.570)$ has eigenvalues $\lambda^s_{1,2} = -480.119, -43.175$ and $\lambda^u = 91.824$. We computed $W^s(\mathbf{x}_s)$ with GlobalizeBVP up to geodesic distance 158.125 when it returns to $\mathbf{x}_s$; the shooting-based method was only able to compute $W^s(\mathbf{x}_s)$ up to geodesic distance 53. The fact that $W^u(\mathbf{x}_s)$ indeed lies on $W^s(\mathbf{x}_s)$ within the resolution of the image demonstrates the accuracy of the computation.

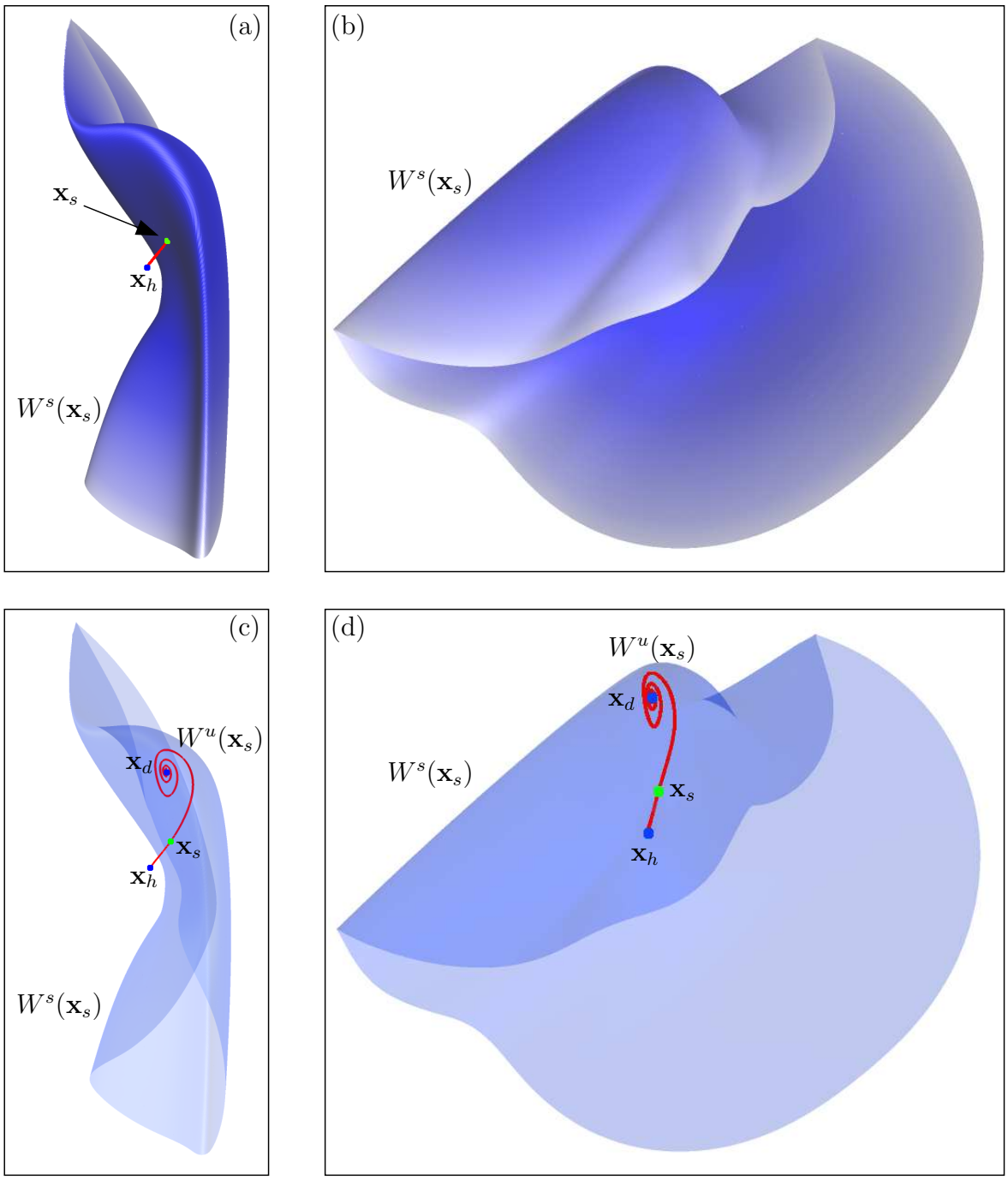

Figure 9: The stable and unstable manifolds, $W^s(\mathbf{x}_s)$ and $W^u(\mathbf{x}_s)$, of the saddle equilibrium $\mathbf{x}_s$ for the somatotroph cell model with $Ca = 0.55$. Panels (c) and (d) show the same views as panels (a) and (b), respectively; $W^s(\mathbf{x}_s)$ is drawn semi-transparent to show $\mathbf{x}_d$ and $W^u(\mathbf{x}_s)$ inside the fold.

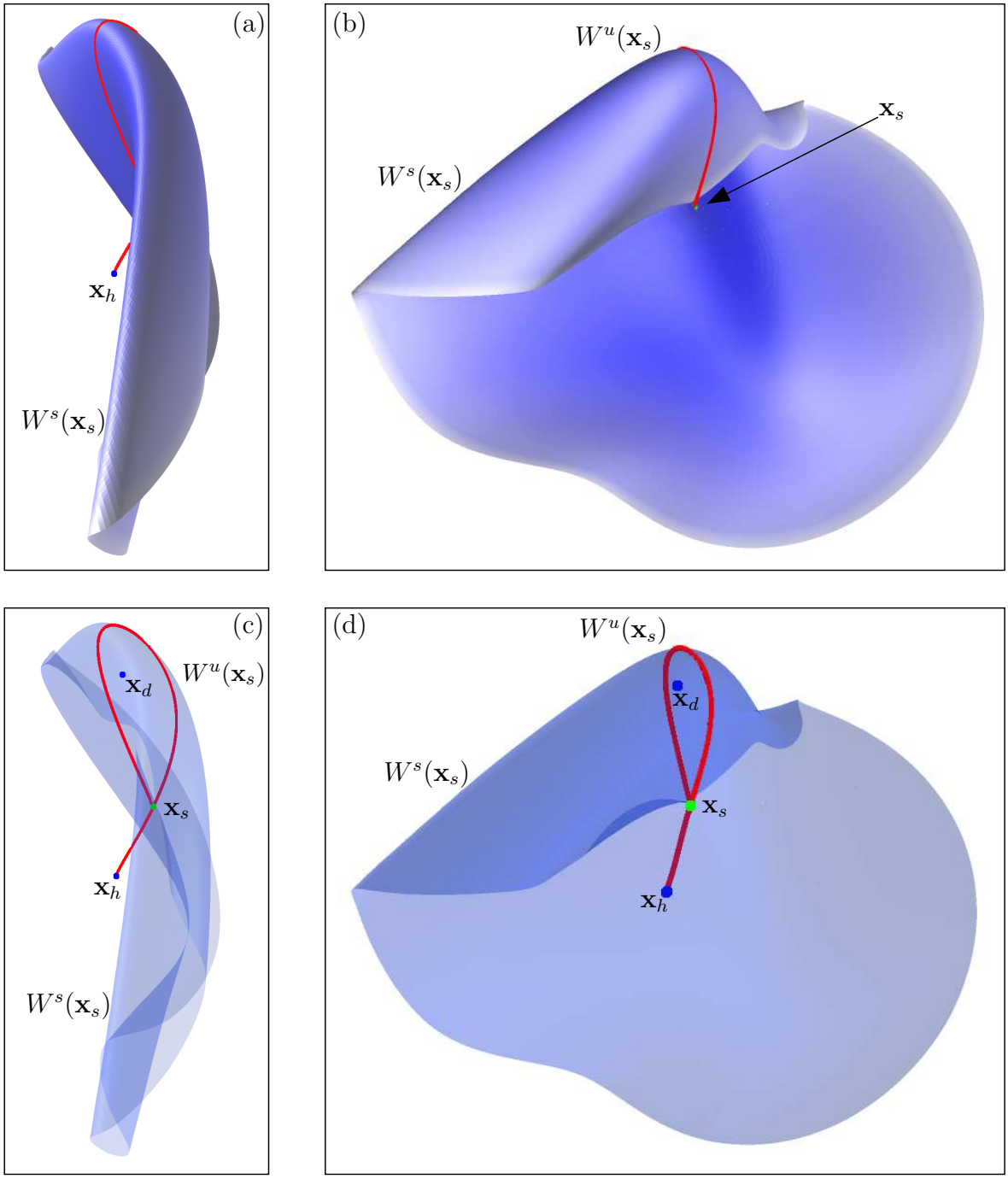

Figure 10: The stable and unstable manifolds, $W^s(\mathbf{x}_s)$ and $W^u(\mathbf{x}_s)$, of the saddle equilibrium $\mathbf{x}_s$ for the somatotroph cell model with $Ca = 0.71672$ where $\mathbf{x}_s$ has a homoclinic orbit. Panels (c) and (d) show the same views as panels (a) and (b), respectively; $W^s(\mathbf{x}_s)$ is drawn semi-transparent to show $\mathbf{x}_d$ and how $W^u(\mathbf{x}_s)$ returns to $\mathbf{x}_s$.

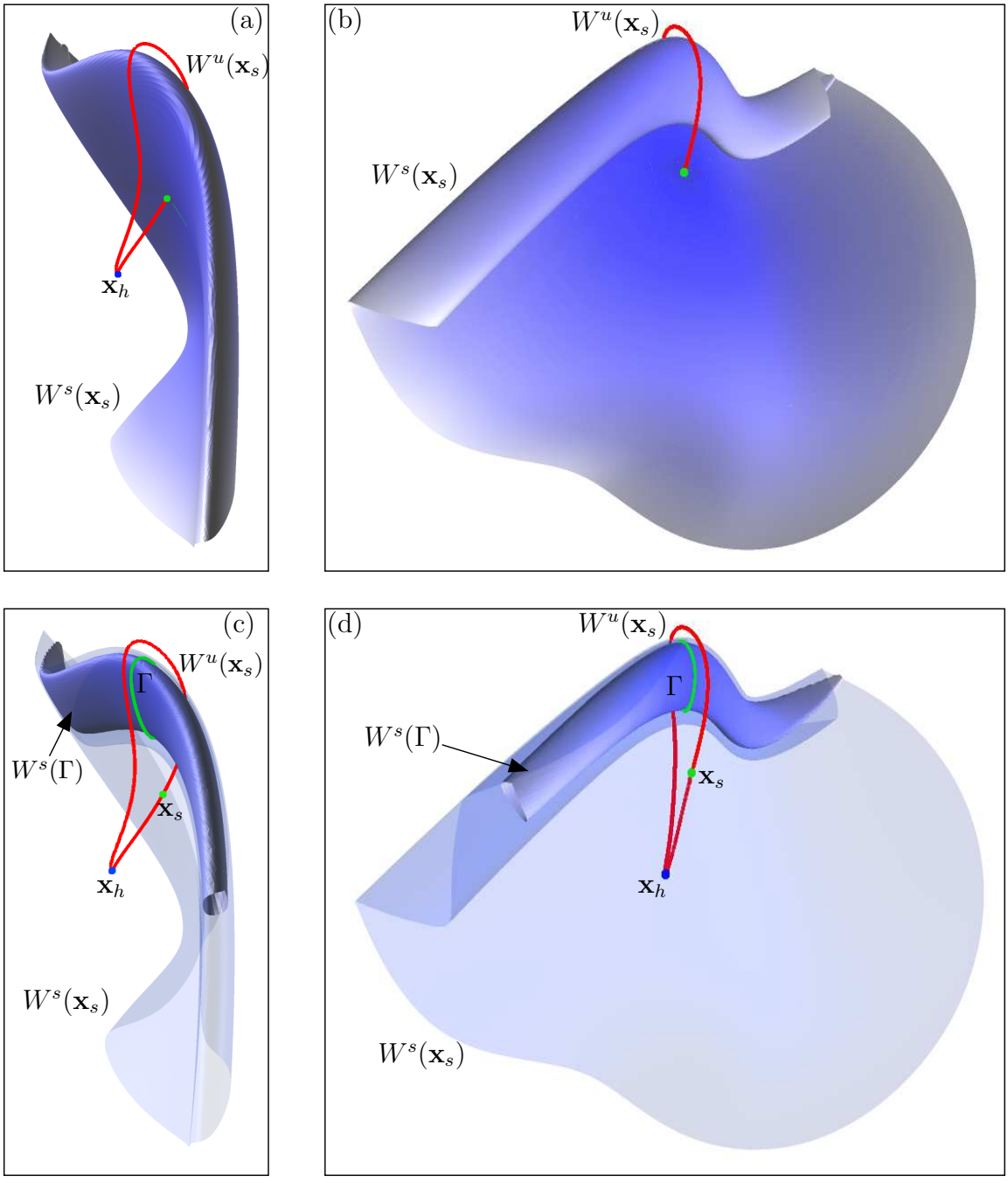

Figure 11: The stable and unstable manifolds, $W^s(\mathbf{x}_s)$ and $W^u(\mathbf{x}_s)$, of the saddle equilibrium $\mathbf{x}_s$ and the stable manifold $W^s(\Gamma)$ of the periodic orbit Γ for the somatotroph cell model with $Ca = 1.0$. Panels (c) and (d) show the same views as panels (a) and (b), respectively; $W^s(\mathbf{x}_s)$ is drawn semi-transparent to show Γ and $W^s(\Gamma)$. Notice how $W^s(\mathbf{x}_s)$ wraps around $W^s(\Gamma)$.

Finally, Figure 11 shows the situation for $Ca = 1.0$, that is, after the homoclinic bifurcation when a saddle periodic Γ exists. We computed the stable manifolds of both $\mathbf{x}_s$ and Γ. Here, $\mathbf{x}_d = (-12.895, 73.278, 48.244)$ and $\mathbf{x}_h = (-62.674, 4.151, 0.106)$; the saddle $\mathbf{x}_s = (-28.935, 41.874, 7.088)$ has eigenvalues $\lambda^s_{1,2} = -528.867, -41.294$ and $\lambda^u = 98.579$. In Figure 11 $W^s(\mathbf{x}_s)$ is computed up to geodesic distance 168, while the left and right branches of $W^s(\Gamma)$ are computed up to geodesic distances 102 and 145, respectively. (We remark that global manifolds of periodic orbits can be computed with GlobalizeBVP in the same was as those of saddle equilibria. The difference lies in the start data that is used; see [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003] for details.) By contrast, the shooting-based method could computed these manifolds only up to respective geodesic distances of 53, 38.6 and 70.8. Notice how the right branch of $W^s(\Gamma)$ in Figure 11(b) and (d) seems to get 'pressed down flat' and 'close up'. This causes meshing problems that indicate the need for an adaptive mesh along the geodesic level sets, which is not implemented at the present. The important difference with Figure 9 for the application is that the unstable manifold $W^u(\mathbf{x}_s)$ now 'escapes' over the folded part of $W^s(\mathbf{x}_s)$, so that both branches of $W^u(\mathbf{x}_s)$ end up at the hyperpolarised state $\mathbf{x}_h$. This situation corresponds to the case shown in Figure 8(c), but there is an important difference. In Figure 8(c) Γ acts as the boundary of the basin of attraction of $\mathbf{x}_d$ and $W^s(\mathbf{x}_s)$ spirals in around Γ. In three dimensions the boundary of the basin of attraction of $\mathbf{x}_d$ is, in fact, the stable manifold $W^s(\Gamma)$ of the periodic orbit Γ; the stable manifold $W^s(\mathbf{x}_s)$ of the equilibrium $\mathbf{x}_s$ 'wraps around' $W^s(\Gamma)$. In other words, if one wants to reach the hyperpolarized state by applying a perturbation then one needs to know the global stable manifold $W^s(\Gamma)$ — knowing only Γ is not enough.

5 Discussion and conclusions

The GlobalizeBVP algorithm presented here uses Auto routines to find and continue solutions of boundary problems needed in the computation of a collection of geodesic level sets as an approximation of a global (un)stable manifold. In this way, it overcomes issues of sensitivity with the previous shooting-based method of [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003]. Furthermore, our method deals successfully with possible tangencies of orbit segments during a computation, which is a challenge for the shooting-based approach. As a result, GlobalizeBVP is able to compute global (un)stable manifolds in difficult circumstances, such as when dealing with a multiple time scale system. As was demonstrated with a slow-fast model of a somatotroph cell, global manifolds can now be computed substantially further than before. This makes GlobalizeBVP a new tool for the study of the global dynamics of slow-fast systems.

Because the accuracy of the boundary value solver can be specified independently, the error control of GlobalizeBVP is the same as for the method of [Krauskopf and Osinga, 1999, Krauskopf and Osinga, 2003]. The user needs to specify the same accuracy parameters before a calculation. Upon reduction of these accuracy parameters the computed manifold converges to the real object in the Hausdorff metric. The computational cost is somewhat larger than for the shooting-based method. However, the difference is smaller than one

might think, because most computation time for the shooting-based method goes into the integration of the vector field, which is performed by collocation in the GlobalizeBVP algorithm.

Having solved one major difficulty, the computation sometimes stops because of problems with the mesh adaptation due to large local curvature. This could be overcome by adjusting the heuristic criteria for adding and removing points during a computation. The further refinements of the meshing strategy are beyond the scope of this paper. We expect that the use of the GlobalizeBVP algorithm for other concrete (slow-fast) systems will give more insight into this issue.

The present implementation is for two-dimensional global manifolds, but the set-up and link with Auto routines works, in principle, for k-dimensional manifolds for any $k \geq 2$. However, representing a three- or even higher-dimensional manifold by an appropriate collection of tetrahedra or higher-dimensional simplices and an associated data structure is a major challenge. Furthermore, the question arises how such a manifold could be visualized to extract useful information from such a computation. Note that, in this respect, it does not differ from other manifold 'growing' techniques, such as those described in [Krauskopf *et al.*, 2005].

6 Acknowledgements

The research of J.E. was supported by grant GR/R94572/01 from the Engineering and Physical Sciences Research Council (EPSRC); B.K. and H.M.O. have been supported by EPSRC Advanced Research Fellowship grants.

References

Doedel, E. J. [1981] "Auto, a program for the automatic bifurcation analysis of autonomous systems," *Congr. Numer.* **30**, 265–384.

Doedel, E. J. , Paffenroth, R. C., Champneys, A. R., Fairgrieve, T. F., Kuznetsov, Yu. A., Oldeman, B. E, Sandstede, B. and Wang, X. J. [2000] Auto2000*: Continuation and bifurcation software for ordinary differential equations*; available via `http://cmvl.cs.concordia.ca/`.

England, J. P., Krauskopf, B. and Osinga, H. M. [2005] "Computing one-dimensional global manifolds of Poincaré maps by continuation," *SIAM J. Appl. Dyn. Syst.* **4**(4), 1008–1041.

Guckenheimer, J. and Worfolk, P. [1993] "Dynamical systems: some compuational problems," in D. Schlomiuk (Ed.), *Bifurcations and Periodic Orbits of Vector Fields*, Kluwer Academic Publishers, pp. 241–277.

Jones, C. K. R. T. [1995] *Geometric singular perturbation theory*, Lecture Notes in Mathematics **1609**, Springer-Verlag, Heidelberg.

Krauskopf, B. and Osinga, H. M. [1999] "Two-dimensional global manifolds of vector fields," *CHAOS* **9**(3), 768–774.

Krauskopf, B. and Osinga, H. M. [2003] "Computing geodesic level sets on global (un)stable manifolds of vector fields," *SIAM J. Appl. Dyn. Sys.* **2**(4), 546–569.

Krauskopf, B., Osinga, H. M., Doedel, E. J., Henderson, M. E., Guckenheimer, J., Vladimirsky, A., Dellnitz, M. and Junge, O. [2005] "A survey of methods for computing (un)stable manifolds of vector fields," *Internat. J. Bifur. Chaos Appl. Sci. Engrg.* **15**(3), 763–791.

Krogh-Madsen, T., L. Glass, L., Doedel, E. J. and Guevara, M. R. [2004] "Apparent discontinuities in the phase-resetting response of cardiac pacemakers," *J. theor. Biol.* **230**, 499–519.

Lorenz, E. N. [1963] "Deterministic nonperiodic flow," *J. Atmos. Sci.* **20**, 130–141.

Osinga, H. M. and England, J. P. [2005] "Separating manifolds in slow-fast systems," in D.H. van Campen, M.D. Lazurko, and W.P.J.M. van den Oever (Eds.), Proceedings of the Fifth EUROMECH Nonlinear Dynamics Conference, ENOC, Eindhoven, 7-12 August, 2005, ID 15-454, 1699–1705.

Osinga, H. M. and Krauskopf, B. [2002] "Visualizing the structure of chaos in the Lorenz system," *Computers & Graphics* **26**(5), 815–823.

Palis, J. and de Melo, W. [1982] *Geometric Theory of Dynamical Systems*, Springer-Verlag, New York/Berlin, 1982.

A. Sherman, A. [1997] "Calcium and membrane potential oscillations in pancreatic beta-cells," in H. G. Othmer, F. R. Adler, M. A. Lewis and J. C. Dallon (Eds.), *Case Studies in Mathematical Modeling - Ecology, Physiology, and Cell biology*, Prentice-Hall, NJ, 1997, pp. 199–217.