England, J., Krauskopf, B., & Osinga, HM. (2003). *Computing one-dimensional stable manifolds of planar maps without the inverse*. https://doi.org/10.1137/030600131

Early version, also known as pre-print

Link to published version (if available):
[10.1137/030600131](10.1137/030600131)

[Link to publication record in Explore Bristol Research](#)
PDF-document

COMPUTING ONE-DIMENSIONAL STABLE MANIFOLDS OF PLANAR MAPS WITHOUT THE INVERSE

J.P. ENGLAND[†‡], B. KRAUSKOPF[†], AND H.M. OSINGA[†]

Abstract. We present an algorithm to compute the one-dimensional stable manifold of a saddle point for a planar map. In contrast to current standard techniques, it is not necessary to know the inverse or approximate it by using Newton's method. Rather than using the inverse, the manifold is grown starting from the linear eigenspace near the saddle point by adding a point that maps back onto an earlier segment of the stable manifold. The performance of the algorithm is compared to other methods using an example where the inverse map is known explicitly. The strengths of our method are illustrated with examples of a stable manifold that is not simply connected and a piecewise-smooth model of an interrupted cutting tool. The algorithm has been implemented for use in the DsTool environment and is available to download with this paper.

Key words. Discrete-time system, planar map, stable manifold, noninvertible, piecewise smooth.

AMS subject classifications. 37D10, 37M20, 65P30.

1. Introduction. Many interesting dynamical systems arising in applications can be described by maps; examples are mechanical systems, lasers, electronic circuits, biological processes and interactions between populations. An important class are Poincaré maps of continuous vector fields, including, in particular, periodically driven systems such as the driven damped pendulum or the forced Van der Pol oscillator; see [4, 20] as general references. Poincaré maps are also often used when the periodicity is intrinsic in the vector field. For such systems the Poincaré map may not be defined globally so that the resulting discrete system is not a global diffeomorphism. Note that there is generally no explicit expression for the Poincaré map of a given vector field. In a number of applications the map is defined explicitly; examples are the Ikeda map [5, 8] and the example of a highly interrupted cutting tool [21] discussed in this paper.

The goal of analyzing a given system is to understand its overall or global behavior. To this end, one needs to find special invariant objects. For two-dimensional discrete-time systems, these are the attractors, the repellors and the saddle points together with their stable and unstable manifolds. In particular, these manifolds tell us a lot about the dynamics of the system. For example, they may form the boundaries of basins of attraction. Furthermore, intersections between stable and unstable manifolds may lead to homoclinic or heteroclinic tangle and chaotic dynamics. Stable and unstable manifolds are global objects that cannot normally be found analytically and, hence, must be computed numerically.

Several methods have been developed to compute the stable and unstable manifolds of a saddle fixed point. We concentrate here on the simplest case: that these manifolds are one-dimensional. Most algorithms use the idea of computing the manifold by starting from a local approximation near the saddle point. A common technique is the concept of iterating a fundamental domain [6, 7, 18, 19, 23]. This involves iterating a local section of the manifold to build up connecting pieces of manifold. An alternative approach, also used here, is to grow the manifold point by point and parameterized by its arclength [10]. All these methods have in common that the stable

[†]Department of Engineering Mathematics, University of Bristol, Bristol BS8 1TR, UK. (james.england@bristol.ac.uk, b.krauskopf@bristol.ac.uk, h.m.osinga@bristol.ac.uk)
[‡]Corresponding author.

manifold is computed as the unstable manifold of the inverse map. This is not a problem when one is dealing with a globally defined Poincaré map. Such a map always has an inverse, which can be computed numerically by following the flow of the vector field backward in time.

However, except in idealized circumstances, an *explicit map* that arises in a particular application does not have an explicit inverse. The expression for such a map is generally quite intricate, contains many parameters and is often written as the composition of several sub-maps. Therefore, it is usually impossible to derive an explicit expression for its inverse. What is more, the map 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. If the inverse exists but is not available in closed form, it is possible to approximate it with Newton's method. To this end, a good seed (initial guess) is required for finding the inverse. Moreover, one needs to know the Jacobian of the map, which may not be well-defined or become singular at certain points. Such requirements are often not practical in applications, as is illustrated with the example of the highly interrupted cutting tool [21] in §4.4.

In this paper we present an 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, our algorithm can also be used in the case where the map is noninvertible, so that multiple pre-images may exist. The algorithm grows the manifold point by point. A new point is found by searching for an intersection point of the image of a circle around the last computed point with the part of the manifold that was already computed. We refer to this as the Search Circle (SC) algorithm. By pre-specifying certain accuracy parameters, a good resolution of the manifold is achieved.

Another computational technique that does not require knowledge of the inverse map is the PIM-triple (Proper Interior Maximum) technique [15], also known as straddling, to compute chaotic saddles. It can be used to find *saddle straddle trajectories* that remain in a given region for a long period of time. If the region contains a saddle point then points on the stable manifold are found because they remain in the region indefinitely. However, the PIM-triple technique has not been used to compute long pieces of stable manifolds.

The SC algorithm is fully implemented in the DsTool environment [1]. The user may download the code for linking with the Tcl/Tk version of DsTool; see [24] for details. All images in this paper were produced with this implementation.

The performance of the SC algorithm is demonstrated with four examples. First we consider a simple test map, the shear map [10], for which the inverse is known explicitly. Secondly, we compute the stable manifold of a modified Ikeda map (without an explicit inverse), which was computed in [9] by approximating the inverse with Newton's method. Our third example is of a map with different numbers of branches of the inverse in different regions of the phase plane. The map was introduced in [14] and we compute parts of the stable manifold, which has infinitely many disjoint branches, for the first time. Our last example is a map of a highly interrupted cutting tool [21]. It does not have an explicit inverse and the map is only piecewise smooth.

The outline of this paper is as follows. First we introduce in §2 the notation and explain some of the mathematical concepts. Then, in §3 we explain the SC algorithm. In §4 we investigate its performance with four examples of planar maps. In §5 we draw conclusions and point to open problems.

2. Mathematical setting. We consider a discrete dynamical system with a two-dimensional state space, given by a continuous map

$$f : \mathbb{R}^2 \mapsto \mathbb{R}^2\,.$$

If f is not orientation preserving we consider its second iterate. We assume that f has a fixed point $x_0 = f(x_0)$ and that f is differentiable in a neighborhood of x_0, but it may not have a unique inverse.

The fixed point x_0 of f is a saddle if the Jacobian matrix $Df(x_0)$ has one stable eigenvalue λ_s inside the unit circle and one unstable eigenvalue λ_u outside the unit circle. The Stable Manifold Theorem [16] guarantees that there exist local stable and unstable manifolds $W^s_{loc}(x_0)$ and $W^u_{loc}(x_0)$ tangent at x_0 to the stable and unstable eigenspaces $E^s(x_0)$ and $E^u(x_0)$, respectively.

The stable manifold $W^s(x_0)$ of x_0 is defined as the set of points that converges to x_0 under forward iteration of f,

$$W^s(x_0) = \left\{ x \in \mathbb{R}^2 : f^n(x) \to x_0 \text{ as } n \to \infty \right\}\,.$$

Similarly, the unstable manifold $W^u(x_0)$ of x_0 consists of points that converge to x_0 under backward iteration of the map f. In terms of forward iterates, this is defined as

$$W^u(x_0) = \left\{ x \in \mathbb{R}^2 : \exists \{q_k\}_{k=0}^{\infty}\,,\ q_0 = x \text{ and } f(q_{k+1}) = q_k \text{ such that } \lim_{k \to \infty} q_k = x_0 \right\}\,.$$

The unstable manifold can also be expressed in terms of the successive union of images of the local unstable manifold $W^u_{loc}(x_0)$, namely

$$W^u(x_0) = \bigcup_{n=1}^{\infty} f^n(W^u_{loc}(x_0))\,.$$

Note that even if f is noninvertible, the images of $W^u_{loc}(x_0)$ will be unique and, hence, the unstable manifold will be simply connected in the phase space.

The global stable manifold $W^s(x_0)$ is similarly defined as the union of the successive pre-images of $W^s_{loc}(x_0)$. However, if multiple inverses exist, then all pre-images, even if disjoint from the main branch, are part of the stable manifold. Hence the stable manifold may or may not be simply connected in phase space.

In §4.3 we examine a noninvertible map from [14]. We find that the stable manifold contains segments with more than one pre-image and is, therefore, no longer simply connected. If this is the case, then there exist curves that separate the phase space into regions with different numbers of pre-images. The stable manifold may cross these curves where the derivative of the map vanishes. Adopting the notation in [14] we define the curves

$$J_0 = \{x \mid Df(x) \text{ is singular}\} \text{ and } J_1 = f(J_0)\,.$$

(Note that in a two-dimensional state space J_0 typically defines a critical *curve*. The image of this curve, J_1, divides the phase space into regions with constant number of pre-images.) The curve J_1 is also called the *Ligne Critique*, or *LC* [3, 14].

3. The Algorithm. The Search Circle (SC) algorithm uses the idea of growing a one-dimensional manifold in steps by adding new points according to the local curvature properties of the manifold. The difference between the SC algorithm and the algorithm from [10] lies in how the next point is found. The algorithm from [10] would use the inverse f^{-1} for finding the next point. The SC algorithm on the other hand, finds a new point close to the last computed point that maps under f to a piece of the manifold that was already computed. Before we explain this in detail, recall for completeness the basic idea of growing the manifold step by step.

3.1. Growing the manifold. The algorithm produces a piecewise linear approximation of $W^s(x_0)$ by computing an ordered list of points $M = \{p_0, p_1, .., p_N\}$ at varying distance from each other. We use a linear approximation for $W^s_{loc}(x_0)$, so 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 as is explained below.

Suppose that the manifold has been grown up to p_k such that $M = \{p_0, p_1, .., p_k\}$. We wish to find the next point p_{k+1} at a distance Δ_k from p_k. The manifold is forward invariant, so new points, and in particular p_{k+1}, must map back onto the piece of the manifold that has been computed so far. Indeed, if f^{-1} is known, then we could search the previously computed part of the manifold to find a point that maps at distance Δ_k from p_k under f^{-1}. This is the method used in [10] to compute the stable manifold. We wish to avoid the use of f^{-1} and use a different method, which is explained in detail in §3.2.

We do use the strategy of [10] to ensure an acceptable resolution of the curve according to pre-specified accuracy parameters. In particular, we monitor α_k, the angle between p_{k-1}, p_k and p_{k+1}, and the product $\Delta_k \alpha_k$. We approximate α_k by

$$\alpha_k = 2\sin^{-1}\left(\frac{\|\bar{p} - p_{k-1}\|}{2\|p_k - p_{k-1}\|}\right) \approx \frac{\|\bar{p} - p_{k-1}\|}{\|p_k - p_{k-1}\|}.$$

where

$$\bar{p} = p_k + \frac{\|p_k - p_{k-1}\|}{\|p_k - p_{k+1}\|}(p_k - p_{k+1})$$

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

$$(3.1) \qquad\qquad\qquad\qquad \alpha_k < \alpha_{max} \quad \text{and}$$
$$(3.2) \qquad\qquad\qquad\qquad \Delta_k \alpha_k < (\Delta\alpha)_{max}.$$

Condition (3.1) ensures that the resolution of the curve is maintained and condition (3.2) 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 Δ_k by $\frac{1}{2}\Delta_k$ and repeat the procedure to find a new candidate for p_{k+1}. We set $\Delta_k = 2\Delta_k$ if both $\alpha_k > \alpha_{min}$ and $\Delta_k \alpha_k > (\Delta\alpha)_{min}$ for a user-specified choice of parameters α_{min} and $(\Delta\alpha)_{min}$. This ensures that the number of points used to approximate the manifold is in some sense optimized for the required accuracy constraints. Note that, at sharp folds it may be necessary to accept $\alpha_k > \alpha_{max}$ due to Δ_k becoming very small. In this case we accept the point if $\Delta_k < \Delta_{min}$, where Δ_{min} is also a predefined parameter, and note the occurrence; see [10] for more details.
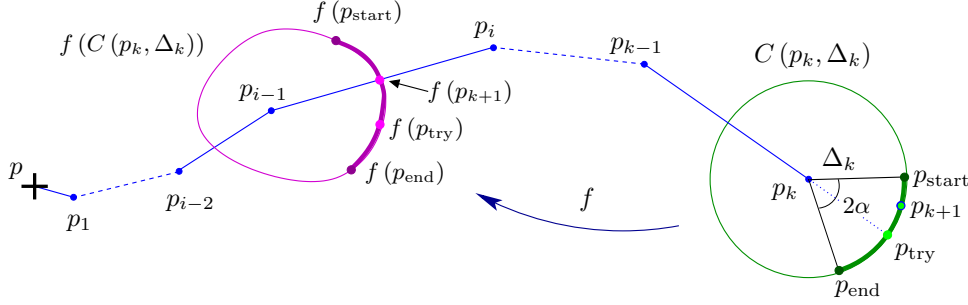
FIG. 3.1. *Graphical illustration of the SC algorithm. A new point p_{k+1} is found on the circle $C(p_k, \Delta_k)$ centered at p_k with radius Δ_k, such that $f(p_{k+1})$ lies on a previously computed part of $W^s(x_0)$.*

3.2. The Search Circle (SC) method. Now, we explain how the next point along $W^s p$ is found when f^{-1} is unknown. The SC algorithm finds the point p_{k+1} at distance Δ_k from p_k that maps back onto a piece of manifold that was already computed. The method is illustrated in Figure 3.1. We know that p_{k+1} must lie somewhere on a circle of radius Δ_k centered at p_k. The image of this circle must then intersect a previously computed part of the manifold. Note that the circle centered at p_k itself necessarily intersects the manifold that has been computed so far. Hence, its image intersects the manifold at the image of this intersection point. If Δ_k is small enough, there will be a unique second intersection, which is the image of the point p_{k+1} that we wish to find. As depicted in Figure 3.1, the point p_{k+1} must lie somewhere on the green circle $C(p_k, \Delta_k)$ in figure 3.1. The image $f(C(p_k, \Delta_k))$ of this circle, is the closed loop shown in magenta. If Δ_k is small enough then $f(C(p_k, \Delta_k))$ will only intersect the manifold twice; one of the intersections we are not interested in as it is the image of a point on the previously computed manifold. The other point intersects the segment, say $[p_{i-1}, p_i]$, of the approximated manifold and it is the image of p_{k+1}, as indicated by the arrow. The point that maps to this intersection is our candidate for p_{k+1}.

The accuracy constraint (3.1) puts an upper bound on the allowable angle between consecutive segments. We can ensure immediately that α_k does not exceed α_{max} by only searching the part of the circle that satisfies this criterion. We denote this search region by $C_\alpha(p_k, \Delta_k)$, which is the arc between p_{start} and p_{end}; see the segment of the green circle in Figure 3.1 enclosed by the black lines. The image of this segment is the thick magenta curve $f(C_\alpha(p_k, \Delta_k))$. Note that this automatically ensures that we do not accidentally search for a pre-image of the other intersection point (that we are not interested in).

In order to find p_{k+1} we need to decide which segment of the calculated manifold contains the intersection point with $f(C_\alpha(p_k, \Delta_k))$. We first try the segment $[p_{i-1}, p_i]$ that was used in the previous step to find a candidate for p_{k+1}. If the image of this candidate lies on the line through p_{i-1} and p_i but not in the segment $[p_{i-1}, p_i]$, we discard this point and repeat the algorithm with the following segment $[p_i, p_{i+1}]$ to find a new candidate for p_{k+1}. (If the map has multiple pre-images then we may need to search for $f(p_{k+1})$ on the previous segment $[p_{i-2}, p_{i-1}]$; see the noninvertible example in §4.3.)
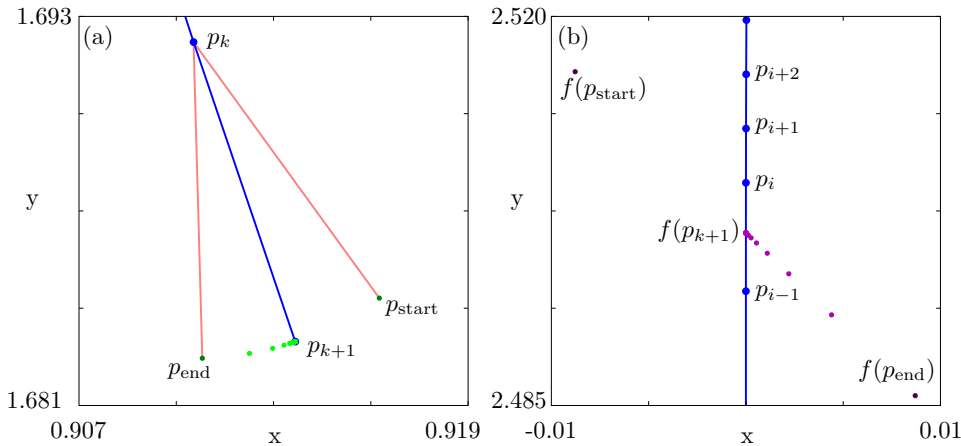
Fig. 3.2. *Computing the stable manifold for the shear map. Searching $C_\alpha(p_k, \Delta_k)$ using bisection to find p_{k+1} (a) and the images of the light green points on $C_\alpha(p_k, \Delta_k)$ in (a) on $f(C_\alpha(p_k, \Delta_k))$ (b). Only $f(p_{k+1})$ also lies on $[p_{i-1}, p_i]$. Therefore, only $p_{k+1} \in C_\alpha(p_k, \Delta_k)$ lies on $W^s(x_0)$.*

We use bisection on $C_\alpha(p_k, \Delta_k)$ to find a candidate for the point p_{k+1} that maps onto the segment $[p_{i-1}, p_i]$. We do not need the point to lie directly on the segment, because this is only an approximation of the manifold. Therefore, we allow the point to lie at a maximum distance of ε_B from the segment $[p_{i-1}, p_i]$, which we call our bisection error; see §3.6 for a discussion on the bisection error. In this procedure we require that p_{start} and p_{end} lie on opposite sides of $[p_{i-1}, p_i]$. Our attempts at finding p_{k+1}, which we call p_{try}, are considered by assessing the sign of $\langle \vec{V}, \vec{W} \rangle$, where $\vec{V} = f(p_{\text{try}}) - p_i$ and $\vec{W}$ is a vector perpendicular to $p_{i-1} - p_i$, such that the product is positive if $f(p_{\text{try}})$ lies on the same side as $f(p_{\text{start}})$ and negative if $f(p_{\text{try}})$ lies on the same side as $f(p_{\text{end}})$. We stop as soon as the normal distance from $f(p_{\text{try}})$ to the segment $[p_{i-1}, p_i]$ is less than ε_B.

Note that, occasionally, p_{start} and p_{end} may not be on opposite sides of the segment $[p_{i-1}, p_i]$. For example, if there is a sharp fold in the manifold, or if the point $f(p_{k+1})$ does not lie in the interval $[p_{i-1}, p_i]$. In the latter case the point will be rejected anyway. Otherwise, we increase α_k and a warning message is given; see [10] for a further discussion of sharp folds.

Once we have found a candidate for p_{k+1} whose image lies within a normal distance ε_B to the line through p_{i-1} and p_i and also in the segment $[p_{i-1}, p_i]$, we then check the accuracy conditions (3.1) and (3.2). If the candidate for p_{k+1} is acceptable then it is added to the list of points M and Δ_k is adjusted if necessary.

Figure 3.2 shows the algorithm in practice, computing the stable manifold for the shear map from (§4.1). Figure 3.2(a) shows the algorithm searching for p_{k+1}, with the points p_{start} and p_{end} in dark green, and the successive points p_{try} of the bisection in light green. Figure 3.2(b) shows the images of these search points, getting closer to the segment $[p_i, p_{i-1}]$. The point p_{k+1} was accepted and the new segment $[p_k, p_{k+1}]$ was added to the approximation of $W^s(p)$.

3.3. Estimating the initial value for Δ_1. When we start our algorithm from the saddle point, the first point p_1 is taken at a small distance δ along the stable eigendirection, $E^s(x_0)$. It is not straightforward what value to choose for Δ_1. This distance must be much smaller than restricted by the accuracy criteria, because the manifold stretches relatively little close to the fixed point. We cannot just iterate the endpoints as we do not know the inverse, but we wish to grow the manifold such that $p_k \approx f(p_{k+1})$.

Hence, our aim is to add a new point p_2 at distance Δ_1 from p_1, such that the point $f(p_2)$ lies on $[x_0, p_1]$ very close to p_1. If we choose Δ_1 too small, then the algorithm computes too many mesh points close to the saddle. However, if we choose Δ_1 too large, then the image of the segment $C_\alpha(p_k, \Delta_k)$ does not intersect $[x_0, p_1]$ at all. A good estimated for Δ_1 is found as follows. The stable eigenvalue λ_s is the linear contraction along the stable eigenvector. Near the saddle, we may use λ_s as an estimate of the nonlinear contraction, so that the initial estimate is

$$\Delta_1 = \delta_1 \left(\frac{1}{\lambda_s} - 1 \right)$$

where $\delta_1 = \delta$ is the distance from x_0 to p_1. Since Δ_1 may still be too large, we search over the interval $\Delta_1 \pm \frac{1}{2}\Delta_1$ on a line through x_0 and p_1 to find an acceptable Δ_1. We use bisection on points in this interval to find points that map close to p_1, assessing which side the points lie of a line perpendicular to the line through x_0 and p_1. We do not have to be exactly on this line but we have to ensure that we do not choose Δ_1 too large. Once we have found an acceptable Δ_1 then the normal SC algorithm is applied. This method may need to be applied several times to find acceptable Δ_k's until the accuracy conditions are met.

3.4. Maps with multiple inverses. For a map with multiple pre-images, the image of $W^s(x_0)$ may cover parts of the manifold several times. In fact, it is possible that the image of one branch of $W^s(x_0)$ covers part of other branch even though the eigenvalues are positive; an example of this phenomenon is investigated in §4.3. In this case the images of points that are added to the manifold traverse back along the computed branch until the fixed point is reached again. That is, the new point that is added to the computed branch is another pre-image of the saddle point, which lies by definition on $W^s(x_0)$. At this moment, further new points map to the other side of the saddle, that is, they lie on the branch that has not been computed yet. Therefore, we must stop and compute this other branch first before the present branch can be grown further. It may be necessary to stop several times and swap between the growth of the two branches to grow one or both up to the required arclength.

In noninvertible maps there may also be disjoint parts of the manifold. The SC algorithm can also be used to find these. This is done by selecting, for example, pre-images of the saddle point that do not lie on the manifold computed so far. The next point is then found without using our α-bounds. Instead we are allowed so search the whole circle for points that map back onto the main branch of the stable manifold. At this initial step we use $\Delta_k = \delta$. Note that when computing a disjoint branch, the SC method can start as soon as *one* point on the manifold has been found and no fundamental domain needs to be constructed; see the example of the noninvertible map in §4.3.
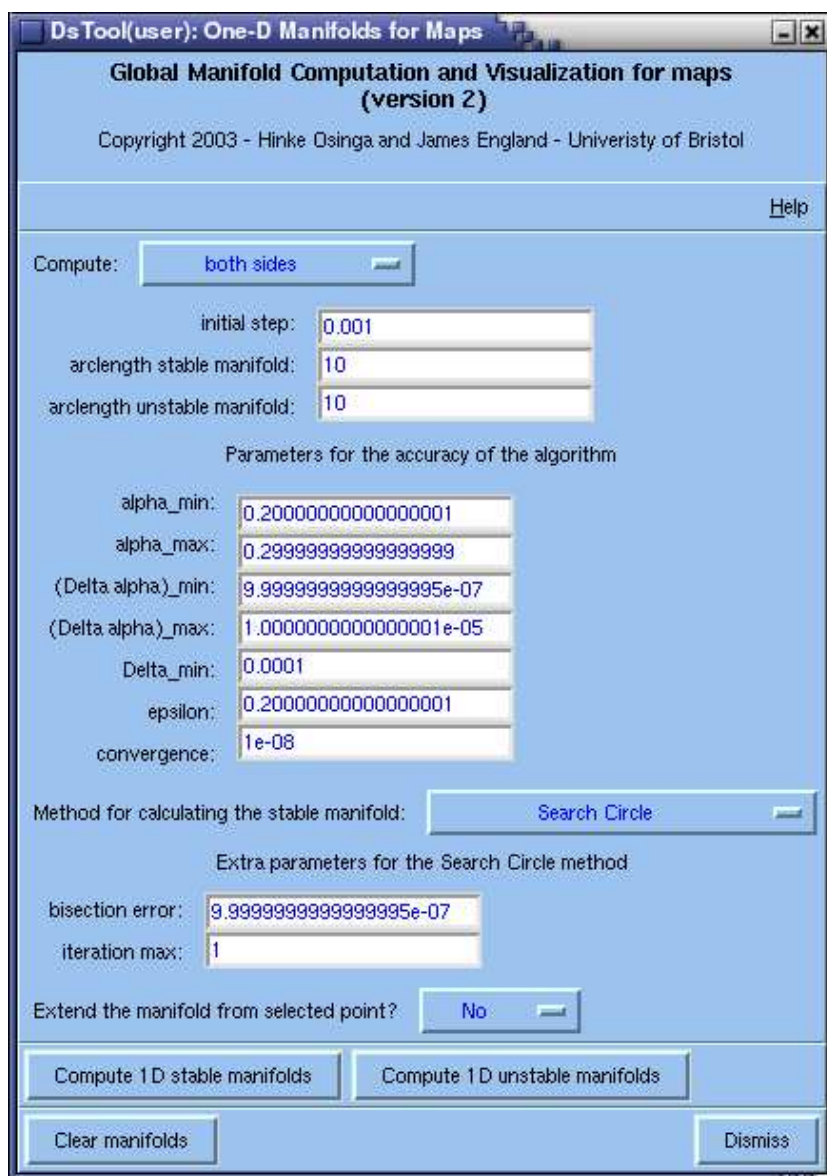
Fig. 3.3. *The implementation of the SC algorithm as it appears in the DsTool environment.*

3.5. DsTool implementation. The SC algorithm has been fully implemented in the Tcl/Tk version of DsTool environment [1]. The code can be downloaded as an extension module to DsTool and is available with this article; see [24]. The installation procedure is straightforward and instructions are included with the code.

Note that our module can be viewed as a replacement of the module presented in [12], which implements the algorithm in [10]. The implementation of the SC algorithm has been combined with that algorithm so that the unstable manifold can be computed with the method of [10] and the stable manifold using either method depending on

whether the inverse is known or not. The window that appears in DsTool when using the module allows the user to choose which method to use to calculate the stable manifold. In fact, it is possible to directly compare the two algorithms in cases where f^{-1} is known or approximated by Newton's method.

A screenshot of the module window in DsTool is shown in Figure 3.3. The window is essentially the same as presented in [12] but with some additional features incorporating the SC algorithm. Firstly, the user can decide which branch to compute: *positive side*, *negative side* or *both sides*, using the selection menu at the top. Below that, one can specify the initial step size δ, and the required arclength of both the stable and unstable manifold computations. Note that it is now possible to only compute, say, the stable manifold without setting the arclength for the unstable manifold to zero. Next follow the accuracy parameters $\alpha_{\max}$, $(\Delta\alpha)_{\min}$, $(\Delta\alpha)_{\max}$, $\Delta_{\min}$, that are used by both algorithms; see §3.1. The parameter ε is the uncertainty factor from [10]. Note that this is not the bisection error ε_B used in the SC algorithm. The parameter *convergence* controls the detection of convergence to a fixed point. This happens when the manifold has finite arclength.

The user is now able to select the method from the drop-down menu that should be used to compute the stable manifold. The options are using an *Explicit/Approximate inverse*, the *Monte Carlo* method and the *Search Circle* method. If *Explicit/Approximate inverse* or *Monte Carlo* method is selected then the algorithm from [10] will be used to compute the stable manifold. The Monte Carlo method uses random seeds for Newton's method to find the inverse, whether an inverse has been defined or not. If no explicit or approximate inverse is defined, then the *Explicit/Approximate inverse* option will not appear in the drop-down menu.

If the *Search Circle* method is selected, then two extra parameters appear. The bisection error ε_B is set in *bisection error*. The parameter *iteration max* is only used when the map has multiple inverses; see §4.3 for more details. The final drop-down menu allows the user to extend the manifold from a selected point. If this option is selected then the algorithm will try and extend disjoint parts of the manifold from points saved in the DsTool *Selected* window. This is important for cases where the map has multiple inverses, such that disjoint branches of the manifold exist. The user must first find a point on a disjoint branch. This can be done, for example, by using Newton's method in combination with Monte Carlo seeds for finding pre-images of the fixed points that do not lie on the branches computed so far.

3.6. Comment on accuracy. A complete discussion of the accuracy of the basic algorithm is given in [6] and [10]. Our method introduces an additional error, which does not appear in [10], due to the bisection error ε_B. This is a parameter of the computation, and we can make it as small as we like. It is not practical to set this parameter to zero, but by setting it to the same order as the interpolation error, the overall error is of the order given in [10].

4. Examples. In this section we use the SC algorithm to compute approximations of the stable manifolds of four planar maps.

The same accuracy parameters were used throughout in the following examples, unless otherwise stated. The initial step is $\delta = 10^{-3}$ and the other accuracy parameters are $\alpha_{min} = 0.2$, $\alpha_{max} = 0.3$, $(\Delta\alpha)_{min} = 10^{-6}$, $(\Delta\alpha)_{max} = 10^{-5}$ and $\Delta_{min} = 10^{-4}$. We chose a bisection error $\varepsilon_B = 10^{-6}$ for all calculations.

We begin with the shear map in §4.1. The shear map is an ideal test case for all algorithms computing stable and unstable manifolds. Namely, the local stable and unstable manifolds are the coordinate axes of the system and there is a homoclinic
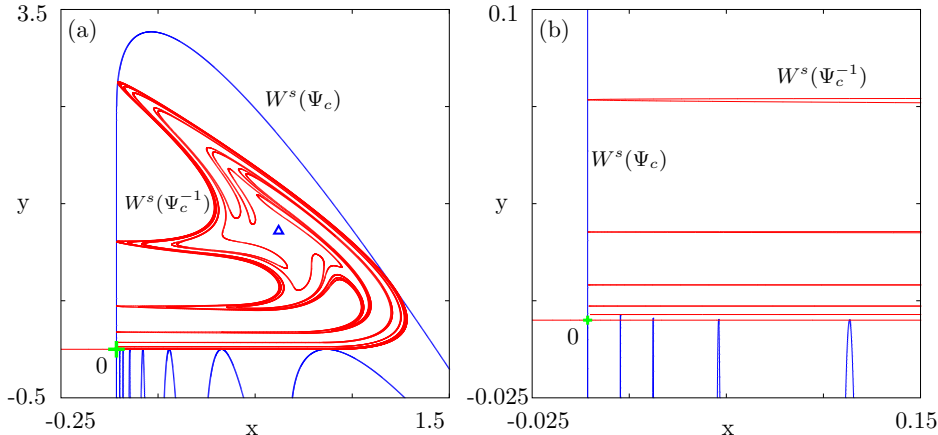
FIG. 4.1. *The Shear map computed at first tangency (c=0.811580) with the stable manifold calculated using the SC method. The unstable manifold was computed as the stable manifold of the inverse of the shear map, also using the SC method. Panel (b) is an enlargement near the fixed point. (The main branch of $W^s(0)$ was computed to arclength 5000 and $W^u(0)$ computed to arclength 100.)*

tangency for a specific parameter value. Therefore, we know that the algorithm is accurate if the folds of the manifolds are tangent to the axes in a neighborhood of the origin.

Our next example, in §4.2, is a modified form of the Ikeda map. It was introduced in [9] as an example of a map that does not have an inverse in closed analytic form. We use this map for comparison to the algorithm in [9], which uses Newton's method to approximate the inverse.

The case of a map with multiple inverses is used in §4.3. Here, the stable manifold consists of many disjoint curves, and our algorithm computes the simply connected piece that contains the fixed point without difficulty; disjoint parts can be computed as well if locations of the respective pre-images of the fixed point are known.

Finally we discuss a model of a highly interrupted cutting tool in §4.4. Most real-life systems are either noninvertible or are hard to invert due to their complexity. For this map, which is the real-life mechanical system from [21] describing a cutting tool, the Jacobian of the system is discontinuous along a straight line in phase space. (The map can be split into two diffeomorphisms each having a unique inverse and the manifolds are continuous across the smoothness boundary.) We compute the stable and unstable manifolds even though they cross the smoothness boundary. To our knowledge, global manifolds of piecewise-smooth systems have not been computed before.

4.1. The shear map. The shear map was first introduced in [10], and is based on an abstract construction given in [17]. The map is such that the origin is always a saddle point and the local stable and unstable manifolds are the coordinate axes of the system.

The shear map is a composition of simple linear saddle-type behavior with a nonlinear shear that is effective only away from a neighborhood of the origin. The
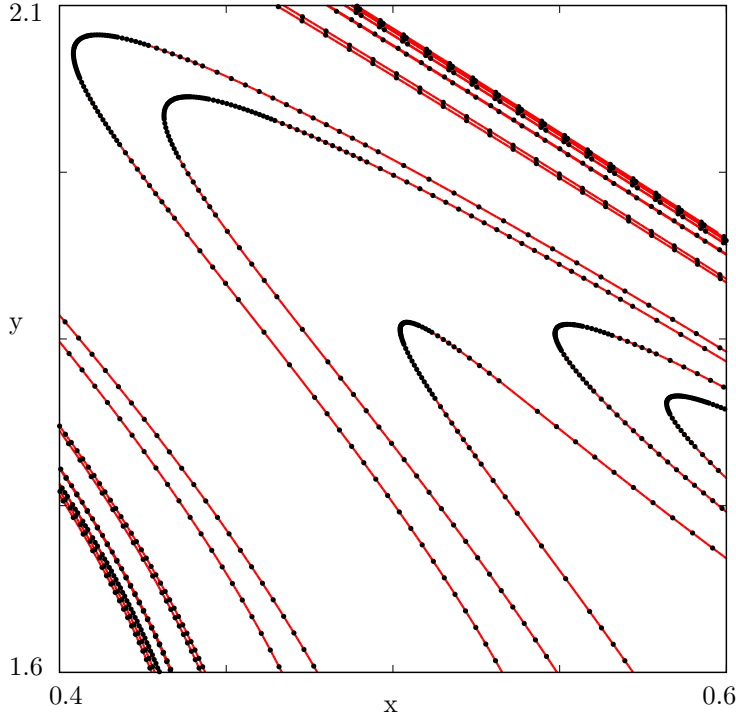
FIG. 4.2. *A part of $W^u(\mathbf{0})$ (red curve) overlaid with black points computed with the SC algorithm as the stable manifold of the inverse map.*

linear map is defined as

$$\phi\left(\begin{array}{c} x \\ y \end{array}\right) = \left(\begin{array}{c} \lambda^u x \\ \lambda^s y \end{array}\right)$$

where the stable and unstable eigenvalues $0 < \lambda^s \leq (\lambda^u)^{-1} < 1$ are fixed parameters. The nonlinear shear ψ_c, depends on the parameter c and is defined as

$$\psi_c\left(\begin{array}{c} x \\ y \end{array}\right) = \left(\begin{array}{c} x - c\, s(x+y) \\ y + c\, s(x+y) \end{array}\right)$$

where,

$$s(z) = \left\{ \begin{array}{cc} 0 & \text{for } z \leq 1, \\ (z-1)^2 & \text{for } z > 1. \end{array} \right.$$

Hence ψ_c produces a quadratic shear along the diagonals $x + y = const > 1$. The shear map is defined as the composition

$$\Psi_c = \psi_c \circ \phi.$$

By construction, $W^s(\mathbf{0})$ contains the half-line $\{x \in (-\infty, 1], y = 0\}$ and $W^u(\mathbf{0})$ contains $\{y \in (-\infty, 1/\lambda^s], x = 0\}$. There is a value $c = c^* \in [0, \lambda^u]$ for which there is
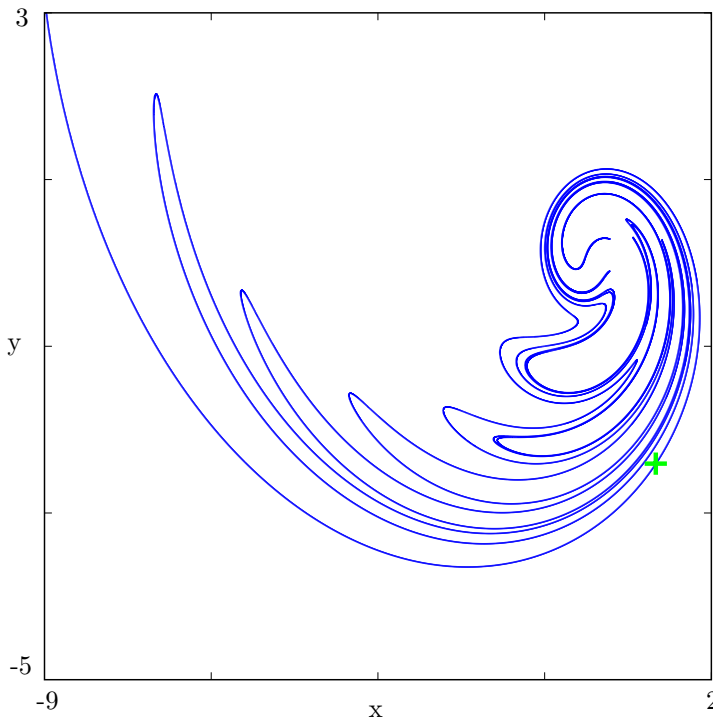
FIG. 4.3. *The stable manifold of the modified Ikeda map up to arclength 150 calculated using the SC method.* $(A = 1, b = 0.9, e = 1, \phi = 0.4, q = 6.)$

a first homoclinic tangency. The homoclinic tangency for $\lambda^u = 0.4$ and $\lambda = 2.0$ was found to be at $c^* \approx 0.811580$, which is precise up to five digits; see [10]. Figure 4.1 shows the stable and unstable manifolds of the shear map both computed using the SC algorithm at this homoclinic tangency. Here Ψ_c was used to calculate $W^s(\mathbf{0})$, shown in blue, and Ψ_c^{-1} was used to calculate $W^u(\mathbf{0})$, shown in red. That is, we calculated $W^u(\mathbf{0})$ as the stable manifold of Ψ_c^{-1} (Note that one would not normally use the inverse to calculate an unstable manifold but we did this for testing purposes because $W^u(\mathbf{0})$ is compact, while $W^s(\mathbf{0})$ is not).

Figure 4.1(b) shows the results of our computation, where we zoomed in near the fixed point, a test that was also performed in [10]. There are no visible differences in the two computations. Note that the limit of numerical accuracy was reached near the fixed point: the last fold of $W^s(\mathbf{0})$ crosses $W^u(\mathbf{0})$, but the last fold of $W^u(\mathbf{0})$ just misses $W^s(\mathbf{0})$.

Figure 4.2 shows in red a close-up piece of $W^u(\mathbf{0})$ calculated using Ψ_c and the algorithm from [10]. The black points are on the stable manifold of Ψ_c^{-1} computed with the SC algorithm. Here we used the larger values $\Delta\alpha)_{min} = 10^{-5}$ and $(\Delta\alpha)_{max} = 10^{-4}$ to demonstrate how the distribution of points is adapted to the curvature. Despite the lower accuracy, the black points computed with the SC method lie exactly on the red manifold.

4.2. Modified Ikeda map. The Ikeda map describes the dynamics of an optical ring laser cavity [5, 8]. A modification of the Ikeda map is used in [9] as an example of a system that does not have a closed analytic form of the inverse map. It is defined by

$$(4.1) \qquad g \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} A + bx \cos m - ey \sin m \\ by \cos m + ex \sin m \end{pmatrix},$$

where

$$m = \phi - \frac{q}{1 + x^2 + y^2}$$

and A, b, e, ϕ and q are parameters of the system. The standard Ikeda map has $b = e$, and it is possible to find an explicit inverse. We consider $b = 0.9$, $e = 1$ and use $A = 1$, $\phi = 0.4$ and $q = 6$ as was done in [9]. Figure 4.3 shows the stable manifold of the saddle $(1.08332, -2.40796)$ indicated by a green cross, computed up to arclength 150. The two branches of the stable manifold are shown in blue; one of the branches immediately goes to infinity; the other branch stretches and folds as it forms a homoclinic tangle creating a chaotic saddle, while at the same time also going to infinity. This branch is computed to approximately the same arclength as shown in [9, Figure 3], where 17 iterates of a fundamental domain were computed and the inverse was approximated by Newton's method. (The exact fundamental domain is not specified in [9].) There appear to be no differences in the approximations of the stable manifold between Figure 4.3 and [9, Figure 3].

4.3. A noninvertible map. A noninvertible map was investigated in [14] to locate pre-images and determine regions of the phase space where there are different numbers of pre-images. The map is given as

$$(4.2) \qquad \Lambda \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x^4 + \nu x^2 + xy + \mu x \\ (1 + a) y + b + \epsilon x^2 \end{pmatrix}.$$

As in [14] we set $\mu = a = b = \epsilon = 0.1$ and $\nu = -1.9$. The map Λ has a saddle fixed point at $p = (0, -b/a) = (0, -1)$ with a negative stable eigenvalue. In [14] the phase space is separated into regions that have different numbers of pre-images by the curve J_1, which is the image of the curve J_0, that defines where the derivative of the map vanishes; see §2. The authors of [14] compute successive pre-images of p to demonstrate this. The stable manifold $W^s(p)$ contains all these pre-images, but the manifold itself is not computed in [14].

The computation of the stable manifold for such a map is difficult because it typically crosses curves where the Jacobian vanishes. Algorithms that use Newton's method, like in [9], will fail to reproduce the complex structure of $W^s(p)$. At best only parts of the connected piece containing p can be computed up to J_0.

We use the built-in algorithm of DsTool for comparison of computing the stable manifold of this map. Figure 4.4(a) shows the result of taking 10 points uniformly distributed on a fundamental domain with $\delta = 10^{-6}$ and iterating these backward 200 times with Newton's method. Since DsTool uses random seeds, the successive pre-images of p already give an idea of what $W^s(p)$ looks like. Also shown is the period-two repellor (red squares) that form the endpoints of the primary branch of $W^s(p)$. Figure 4.4(b) by comparison shows all points that map to the saddle after 15 iterations, as in [14]. The number of iterations needed to reach the saddle is indicated
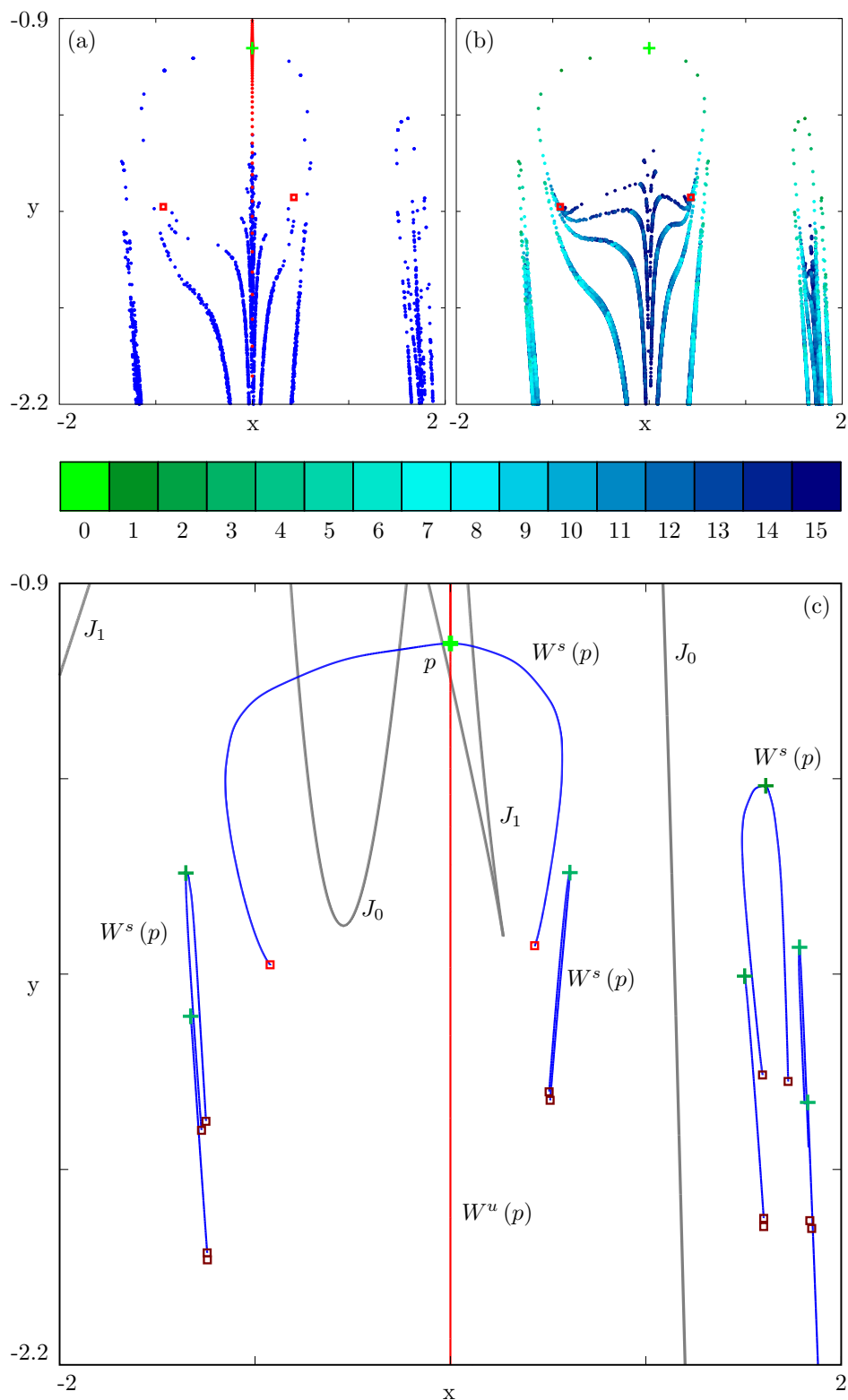
FIG. 4.4. *The stable (blue) and unstable (red) manifolds of the map defined in (4.2) with $\mu = a = b = \epsilon = 0.1$ and $\nu = -1.9$. The red squares in panels (a), (b) and the corresponding squares in panel (c) are a period-two repellor. The other squares in panel (c) are pre-images of this period-two repellor. Panel (a) shows the manifolds computed using the built-in method of DsTool and panel (b) shows all points that map onto the saddle after at most 15 iterations. The color of the points indicates the number of iterations it takes to map to the saddle point; see the color bar. Panel (c) shows the stable manifold computed using the SC method up to arclength 10. (The unstable manifold is comuted up to arclength 10, shown in red.) Note how the manifold is extended from selected pre-images of the saddle point. All points on the disjoint branches of manifold map to the primary branch within 12 iterations.*
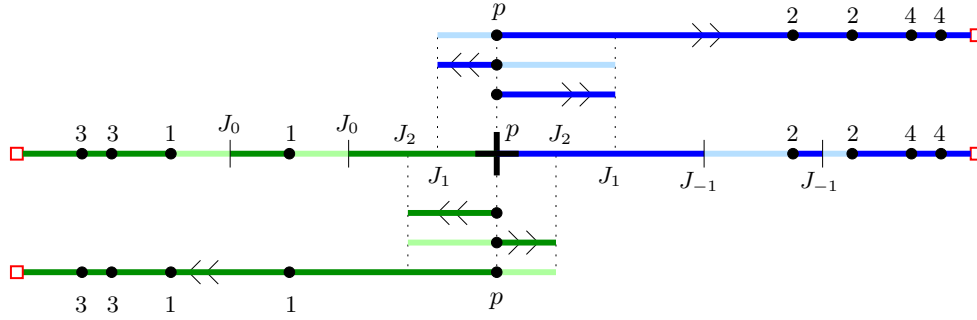
FIG. 4.5. *Schematic illustration of how the image of the primary branch of the stable manifold $W^s(p)$ under Λ^2 covers itself. The red squares are a period-two repellor. Segments of a particular color map to similarly colored segments, whereby points marked 3 and 4 are mapped to 1 and 2, respectively, and 1 and 2 map to the saddle p.*

by color, ranging from green (1 iterate) to blue (15 iterates). This color coding is indicated in the color bar and it is maintained throughout the other figures in this example.

Figure 4.4(c) shows, in blue, parts of the stable manifold computed using the SC algorithm. Note that the computations are done using Λ^2 to preserve the orientation. The unstable manifold is shown in red and was computed using the algorithm in [10]. This manifold is the line $x = 0$. The primary branch of $W^s(p)$, namely the simply connected curve containing p, ends at a period-two repellor and contains a number of pre-images that map directly onto the saddle point. The disjoint parts of the manifold are also computed using the SC algorithm. Indeed they are grown by starting at pre-images of p indicated by the green crosses in the Figure 4.4(c); the necessary modification of our algorithm is discussed later. The curves J_0 and J_1 are shown in gray. Inside the wedge formed by J_1 the map Λ has four pre-images, whereas outside it there are at most two. Hence, the primary branch of $W^s(p)$ consists both of points with four and with two pre-images.

Figure 4.5 gives a schematic representation of the the primary branch and its image under Λ^2. The primary branch with the period-two repellors as its end points is shown as a straight line. The saddle point p is the cross in the center. The pre-images of p are marked in the figure by the number of iterations of Λ that it takes to map to p, up to Λ^4. Also marked are the intersections with the curves J_{-1}, J_0, J_1 and J_2, where J_{-1} is the pre-image of J_0 (effectively J_{-1} is part of the J_0 curve of Λ^2) and $J_2 = \Lambda(J_1) = \Lambda^2(J_0)$. Consecutive sections on the primary branch are indicated by alternating colors, green for the left branch and blue for the right branch. Correspondingly colored pieces of the Λ^2-image of the manifold are shown above and below the primary branch.

To help interpret this figure, let us start by examining the right-hand side of the primary branch. The SC algorithm grows the manifold starting from a fundamental domain with $\delta = 10^{-3}$. The segment from the saddle point up to its first intersection with J_{-1} is mapped one-to-one to the segment up to the intersection with J_1, which is indicated by the similarly colored piece of the manifold shown immediately above the primary branch. The next segment from the intersection with J_{-1} to the point marked 2, indicated by the lighter color, is then mapped back to the piece of manifold
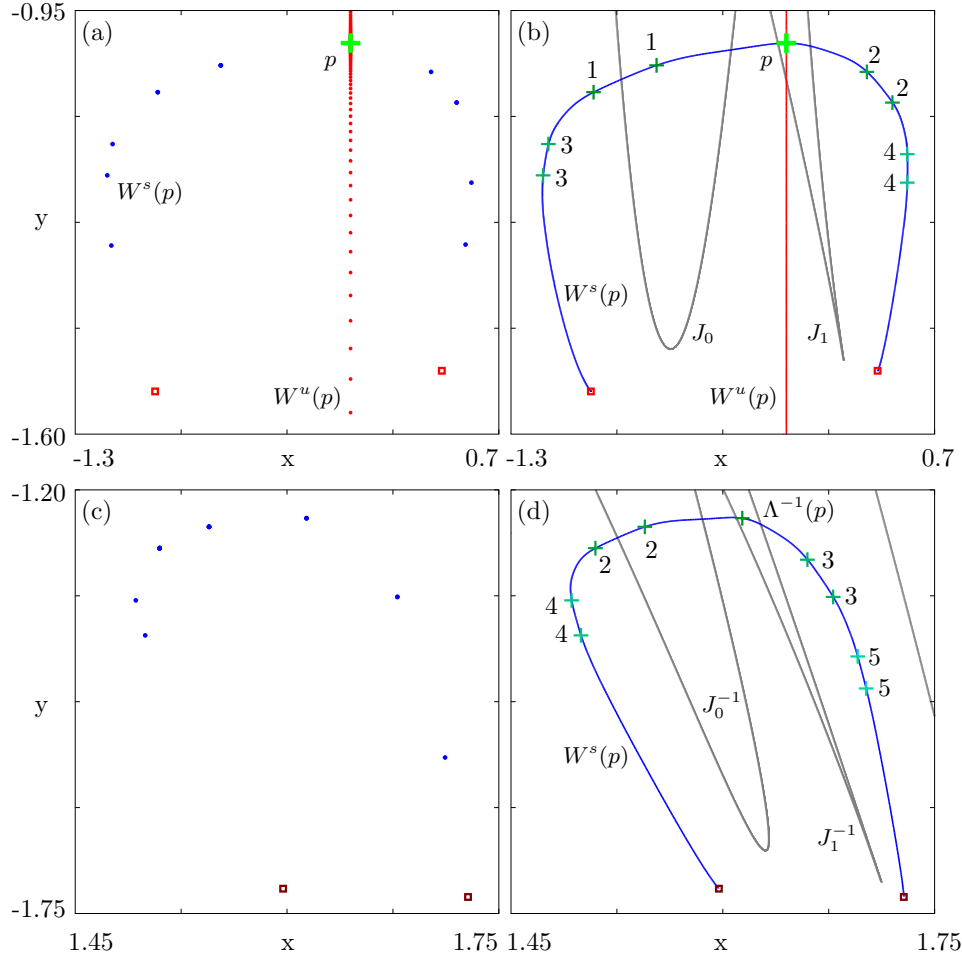
FIG. 4.6. *The stable manifold of the map defined in (4.2) with $\mu = a = b = \epsilon = 0.1$ and $\nu = -1.9$. Panels (a) and (c) show points on the primary branch, and a pre-image of the primary branch computed using the built-in method of DsTool. Panels (b) and (d) show the primary branch and a pre-image of the primary branch computed using the SC method. The color of the crosses is as in Figure 4.4 indicated by the number of iterations needed for the point to map to the saddle point.*

that traverses back from J_1 to the saddle point p.

The segment between the point marked 2 up until the next point marked 2 maps to the other side of the saddle. Our algorithm stops as we have not yet found this piece of manifold and so we cannot check if points map onto it. We would have to calculate the branch of the manifold on the other side of the fixed point and swap back between the two sides of manifold to grow it further. For the map Λ, we get around this problem by using a trick: we try one further iteration of the points to see if they lie on a piece of the manifold already calculated on the same branch. We can see from Figure 4.5 that the segment between the two points marked 2 on the

manifold maps to segments of manifold on the other side of the saddle under Λ^2. The section would map back to the right side of the manifold under Λ^3, which is a segment of manifold we have already computed.

The segment containing the two points marked 4 map to the entire right branch under Λ^2. The image of the left branch folds similarly, covering part of the right branch up to $J_2 = \Lambda(J_1)$.

In [14] it was observed that the set of pre-images appears to contain teardrop-shaped clusters that are mapped onto each other. We compute other disjoint branches of $W^s(p)$ by selecting pre-images of the saddle point that lie on these teardrop-shaped branches. These points are shown in Figure 4.4(c) as green crosses (color coded relative to the number of iterations it takes to map to p). In the next step we apply the SC algorithm to find a new point that maps to the primary branch of the manifold. We can now continue along any branch of manifold found mapping back to the primary branch. As was done earlier, it is sometimes necessary to change the number of iterates of Λ that are used.

In Figure 4.6 we zoom into the primary branch and a pre-image of the primary branch of $W^s(p)$. Figures 4.6(a) and 4.6(c) show the primary branch and a pre-image of the primary branch, computed using Newton's method with the built-in algorithm of DsTool (again only 10 points on a fundamental domain were iterated backward 200 times). Figures 4.6(b) and 4.6(d) show the same branches of $W^s(p)$ computed using the SC method. Overlaid on the branches are pre-images of the fixed point indicated by a colored cross and labeled according to the number of iterations it takes to map back to p. The endpoints of these branches end up at pre-images of the period-two repellor but in themselves they are not period-two points. We can see how pre-images of p on the secondary branch map to pre-images on the primary branch.

In Figure 4.6(d) we labeled the pre-images of the J_0 and J_1 curves as follows. We define $J_0^{-1} \supseteq J_{-1}$ such that $\Lambda(J_0^{-1}) = J_0$ and $J_1^{-1} \supseteq J_0$ such that $\Lambda(J_1^{-1}) = J_1$.

Note that there certainly exists a pre-image of the unstable manifold in Figures 4.6(c) & (d) but this pre-image is not part of the unstable manifold, because the manifold in only forward invariant and is connected to the saddle point.

4.4. The highly interrupted cutting map. We consider a piecewise smooth map given in [21] that models a cutting tool. The tool oscillates with a fixed frequency of period τ. This period is made up of two parts, $\tau_1 + \tau_2 = \tau$. The tool oscillates freely for time τ_1 when the tool is not in contact with the workpiece and the machine re-adjusts its position depending on the thickness of the previous cut. The tool then cuts for time τ_2, which is small compared to τ. The map describes the changes in the height x_j of the cutting tool and its velocity v_j at the moment just after the cutting time τ_2. The map is defined as

$$(4.3) \qquad \begin{cases} x\,(t_{j+1}) = A_{11}x\,(t_j) + A_{12}v\,(t_j)\,, \\ v\,(t_{j+1}) = A_{21}x\,(t_j) + A_{22}v\,(t_j) + F(x(t_j),(v(t_j))\,, \end{cases}$$

where A_{11}, A_{12}, A_{21} and A_{22} are functions of the period τ and the relative damping factor ζ, namely

$$\begin{aligned} A_{11} &= \tfrac{e^{-\zeta\tau}}{(1-\zeta^2)}\left(\cos\left(\sqrt{(1-\zeta^2)}\tau\right) + \zeta\sin\left(\sqrt{(1-\zeta^2)}\tau\right)\right), \\ A_{12} &= \tfrac{e^{-\zeta\tau}}{(1-\zeta^2)}\sin\left(\sqrt{(1-\zeta^2)}\tau\right), \\ A_{21} &= -\tfrac{e^{-\zeta\tau}}{(1-\zeta^2)}\sin\left(\sqrt{(1-\zeta^2)}\tau\right), \\ A_{22} &= \tfrac{e^{-\zeta\tau}}{(1-\zeta^2)}\left(\cos\left(\sqrt{(1-\zeta^2)}\tau\right) - \zeta\sin\left(\sqrt{(1-\zeta^2)}\tau\right)\right). \end{aligned}$$
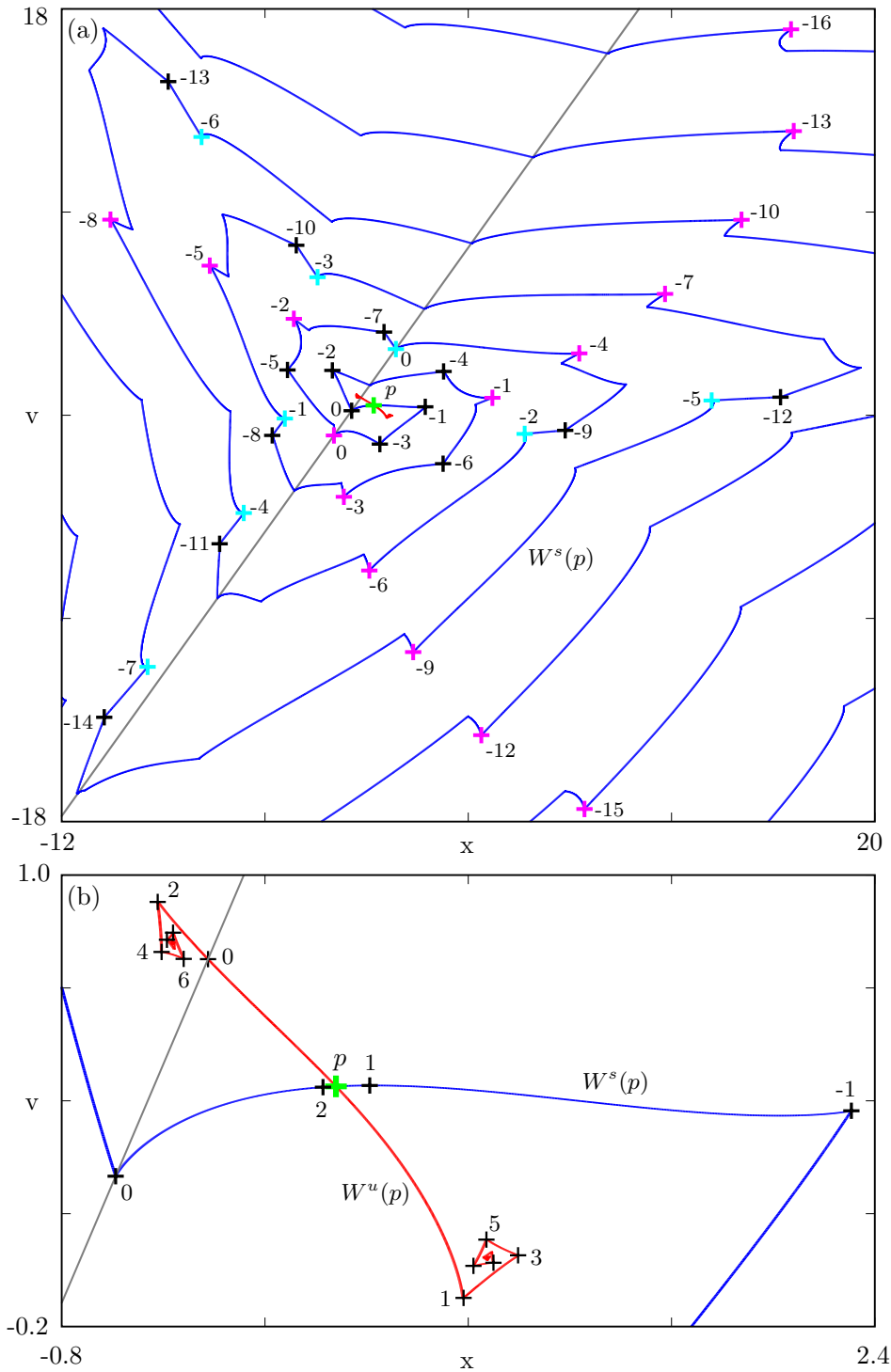
FIG. 4.7. *Panel (a) shows the stable (blue) and unstable (red) manifolds of the saddle p of the map defined in (4.3) for $\zeta = 0.01$, $\tau = 2$, $h_0 = 1$, $K_2 = 0.87$. We computed the stable manifold up to arclength 400 and the unstable manifold up to arclength 10. The gray line indicates the smoothness boundary. Crosses with the same colors are part of the same orbit on the stable manifold, which indicates how sharp corners on the manifold map to intersections with the smoothness boundary. An enlargement of $W^u(p)$ is shown in panel (b); the black crosses on the unstable manifold are images of the intersection point with the smoothness boundary. This clearly shows how sharp corners emanate from the smoothness boundary.*

The function F in the equation for $v(t_{j+1})$ is defined as

$$F(x(t_j), v(t_j)) = \begin{cases} \frac{\tau_2}{m\omega_n^2} F_c(h), & \text{if } h = h_0 + x(t_j) - A_{11}x(t_j) - A_{12}v(t_j) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Here, m is the mass of the tool, h_0 is the feed during one cutting period and ω_n is the natural damping frequency. The cutting force $F_c(h)$ is estimated by the *three-quarters rule*, an empirical formula, giving $F_c(h) = Kwh^{3/4}$ for some parameter K, where w is the chip width and h defines the cutting depth. This term is only included for $h = h_0 - A_{11}x(t_j) - A_{12}v(t_j) + x(t_j) > 0$, that is when the tool cuts the material, and is zero otherwise; see [21] for more details on the derivation of this model. We combine the constants τ_2, m, ω_n, w and the parameter K giving a new parameter K_2. The equation $h = 0$ defines a straight line in (x,v)-space along which the system is not differentiable. The map (4.3) is complicated and finding an explicit inverse is impractical.

Figure 4.7 shows the piecewise smooth manifolds of the map defined in (4.3). For our computation we chose $\zeta = 0.01$, $\tau = 2$ and $h_0 = 1$. For $K_2 = 0.87$ there is a saddle point p at $x_e = 0.27932$, $v_e = 0.43835$, with two negative eigenvalues. This is indicated by a green cross in the figure. Figure 4.7(a) shows the two branches of the stable manifold in blue computed up to arclength 400 using the SC method for the second iterate of the map. The unstable manifold, shown in red, is computed up to arclength 10 using the algorithm from [10].

The branches of the stable manifold spiral outward with obvious non-smooth corners. The branches of the unstable manifold also have non-smooth corners as they approach the period-two repellor; see the enlargement in Figure 4.7(b). The gray line shows the smoothness boundary (switching manifold) along which $h = 0$.

The fixed point is not on the boundary, so in a neighborhood of the saddle the system is smooth and smooth local manifolds exist. The global stable manifold is non-smooth at the point where it intersects the smoothness boundary, but there are also non-smooth points not on this boundary. These non-smooth points are pre-images of intersections with the smoothness boundary, so the non-smoothness propagates backward along the manifold. Hence, the stable manifold is smooth until it arrives at the smoothness boundary or a pre-image of the smoothness boundary. To illustrate this, some of the intersections with the boundary are marked by a colored cross and a zero in Figure 4.7. Points that map to these intersection points are indicated by similarly colored crosses labeled with a number indicating the number of iterations it takes to map to the first intersection point.

Figure 4.7(b) shows a close-up of the unstable manifold. The unstable manifold was computed with the algorithm in [10], and accumulates on a period-two attractor. We see that the intersection of the unstable manifold with the smoothness boundary are smooth, but sharp corners propagate forward along the manifold from this intersection point.

In [21] the authors perform a bifurcation analysis on the parameter K. We can similarly vary the parameter K_2. For $K_2^* = 0.856468$ there is a sub-critical period-doubling bifurcation. Here, the fixed point becomes stable and a period-two saddle emerges between the fixed point and the period-two attractor. In contrast to the period-two attractor, both points on the period-two saddle lie on the same side of the smoothness boundary. In Figure 4.8 we computed the stable manifolds (blue) and the unstable manifolds (red) of the two period-two saddle points for $K_2 = 0.85$. The period-two saddles are indicated by the green crosses and the attractors by blue
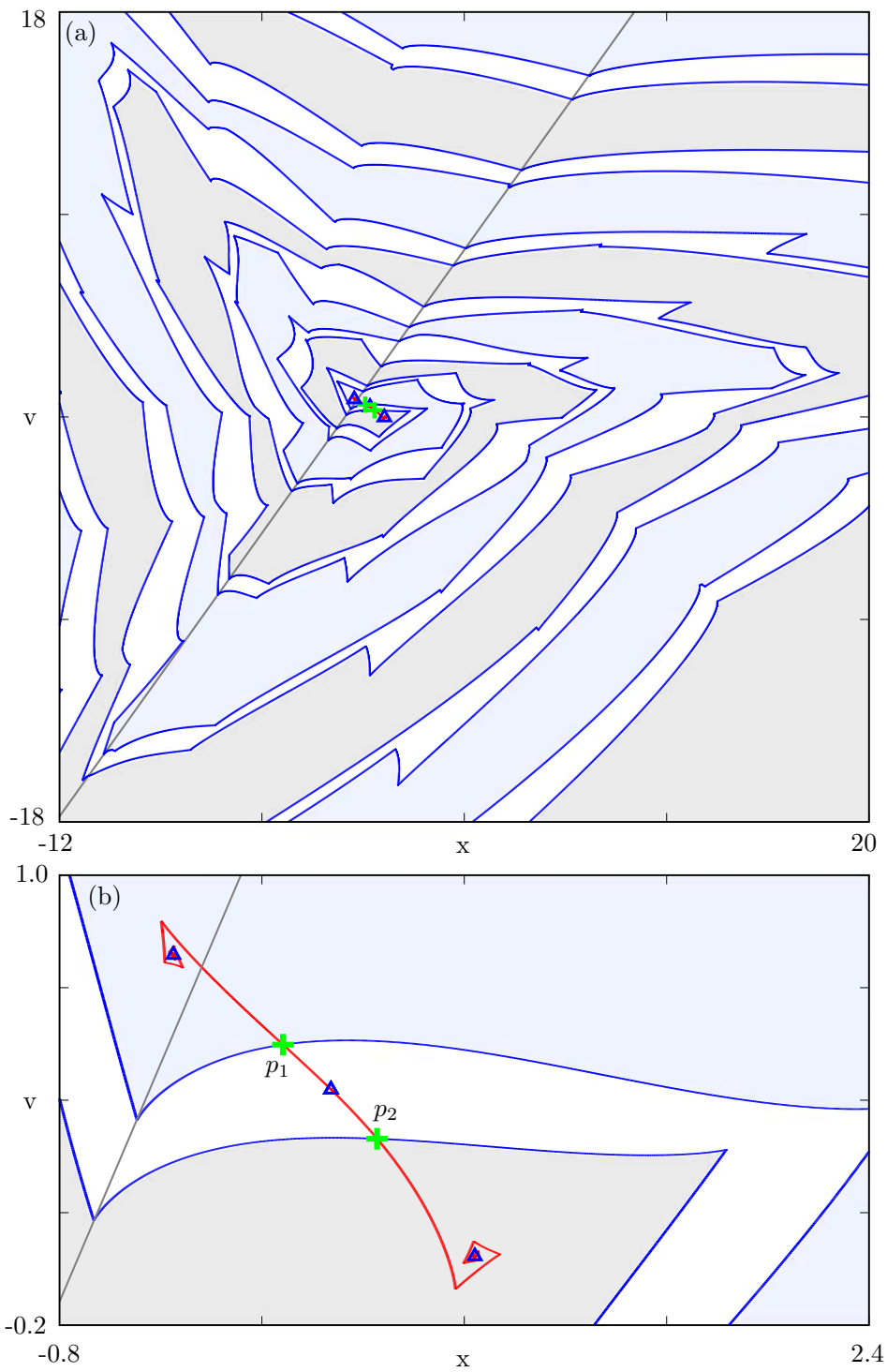
Fig. 4.8. *The stable (blue) and unstable (red) manifolds (a) of a period-two saddle of the map defined in (4.3) for $\zeta = 0.01$, $\tau = 2$, $h_0 = 1$, $K_2 = 0.85$. Again, the stable manifolds are computed up to arclength 400 and the unstable manifolds up to arclength 10. The gray line indicates the smoothness boundary. An enlargements of $W^u(p)$ is shown in panel (b).*

triangles. The stable manifolds produced using the SC algorithm in this case define the boundaries of the basins of attraction between the period-two attractor and the attracting fixed point. Points in the white region tend to the attracting fixed point, whereas points in the shaded regions oscillate between the blue and gray regions tending to the period-two attractor. A more detailed study of this map will be presented elsewhere.

5. Conclusions and Discussion. Many real-life systems are truly noninvertible or very hard to invert, so the inverse is not known explicitly. The SC algorithm described in this paper computes the one-dimensional stable manifold for a planar map without requiring knowledge of the inverse map or the Jacobian matrix. Even if the inverse map is known explicitly, the SC algorithm can still be used to compute the stable manifold.

Until now, stable manifolds of systems with multiple pre-images could not be computed past points where the Jacobian is singular. We have shown that the SC algorithm is not affected by this problem. Furthermore, the SC algorithm can also be used to compute disjoint branches of the stable manifold. One only requires knowledge of the pre-images of the fixed point. Finally, we have shown that our algorithm is unaffected in systems where the derivative of the system is not continuous. To our knowledge, manifolds of such piecewise-smooth systems have not previously been computed. We believe that these properties of the algorithm are very useful when one is exploring real-life systems.

The SC algorithm has been implemented to compute one-dimensional stable manifolds of planar maps. Although planar maps (already) display many types of dynamical behavior including chaos, interesting dynamics can also be investigated in higher dimensions. Let us discuss the implications of generalizing the SC algorithm to higher dimensions. To compute a two-dimensional stable manifold in a three-dimensional phase space, we could use a direct combination of [11] and the SC algorithm. In fact, any system with codimension-one stable manifolds can be computed by generalizing the method in [11] and using it in combination with the SC algorithm. The complexity of the algorithm is of the same order as the method in [11] for dimensions greater than two.

As mentioned in the introduction, the PIM-triple procedure of [15], designed for finding chaotic saddles can be used to calculate stable manifolds of saddle points. The PIM-simplex method in [13] is an extension of the PIM-triple method in [15] for computing chaotic saddles having unstable dimension two or higher. This method uses a non-degenerate simplex to probe the space replacing the line segment; see also [22]. We can generalize our algorithm in a similar way. Instead of searching over a segment of a circle in two dimensions for our next point, we can search over the surface of an n-dimensional hypersphere in n dimensions. However, computing only one-dimensional manifolds of an n-dimensional system would lead to an increased complexity that is not reflected in the dimension of the manifold. For these cases, knowledge of the inverse would dramatically speed up the computation time.

We expect the SC algorithm to be useful in helping to understand the dynamics of many real-life noninvertible and/or non-smooth systems. The great advantage of this method is that only a minimal amount of knowledge of the system is required to compute the stable manifold. Extensions of this algorithm to higher dimensions is left for future work.

REFERENCES

[1] A. Back, J. Guckenheimer, M. R. Myers, F. J. Wicklin, and P. A. Worfolk, *DsTool: Computer assisted exploration of dynamical systems*, Notices Amer. Math. Soc. Vol 39, (1992) pp. 303.

[2] M. Dellnitz and A. Hohmann, *A subdivision algorithm for the computation of unstable manifolds and global attractors*, Numerische Mathematik, Vol 75 (1997) pp. 293–317.

[3] C. E. Frouzakis, L. Gardini, I. G. Kevrekidis, G. Millerioux and C. Mira, *On some properties of of invariant sets of two-dimensional noninvertible maps*, Int. J. Bifurcation and Chaos, Vol 7, No. 6 (1997) pp. 1167–1194.

[4] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, Springer-Verlag, New York/Berlin, 1983.

[5] S. M. Hammel, C. K. R. T. Jones and J. V. Maloney, *Global dynamical behaviour of the optical field in a ring cavity*, J. Opt. Soc. Am. B 2 (1985) pp. 552–564.

[6] D. Hobson, *An efficient method for computing invariant manifolds*, Journal of Computational Physics, Vol 104 (1991) pp. 14–22.

[7] A. J. Homburg, H. M. Osinga and G. Vegter, *On the computation of invariant manifolds of fixed points*, Zertschrift für angewandte Mathematik und Physik, Vol 46 (1995) pp. 171–187.

[8] K. Ikeda, *Multiple-valued stationary state and its instability of the transmitted light from a ring cavity system*, Optics Communications, Vol 30 (1979) pp. 257–261.

[9] E. J. Kostelich and J. A. Yorke and Z. You, *Plotting stable manifolds: error estimates and noninvertible maps*, Physica D, Vol 93 (1996) pp. 210–222.

[10] B. Krauskopf and H. M. Osinga, *Growing 1D and quasi-2D unstable manifolds of maps*, Journal of Computational Physics, Vol 146 (1998) pp. 406–419.

[11] B. Krauskopf and H. M. Osinga, *Globalizing two-dimensional unstable manifolds of maps*, Int. J. Bifurcation and Chaos, Vol 8, No. 3 (1998) pp. 483–503.

[12] B. Krauskopf and H. M. Osinga, *Investigating torus bifurcations in the forced Van der Pol oscillator*, in Book Title: Numerical Methods for Bifurcation Problems and Large-Scale Dynamical systems, edited by E. J. Doedel, B. Fielder, I. G. Kevrekidis, W.-J. Beyn, J. Lorenz, L. Tuckerman, E. S. Titi, H. B. Keller, and D. G. Aronson, IMA Volumes in Mathematics and its Applications, Springer-Verlag, New York, 1999.

[13] P. Moresco and S. P. Dawson, *The PIM-simplex method: an extension of the PIM-triple method to saddles with an arbitrary number of expanding direction*, Physica D 126 (1999) pp. 38–48.

[14] Chia-Hsing Nien and F. J. Wicklin, *An algorithm for the computation of preimages in noninvertible mappings*, Int. J. Bifurcation and Chaos, Vol 8, No. 2 (1998) pp. 415–422.

[15] H. E. Nusse and J. A. Yorke, *A procedure for finding numerical trajectories in chaotic saddles*, Physica D 36, (1989) pp. 137-156.

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

[17] J. Palis and F. Takens, *Hyperbolicity and Sensitive Chaotic Dynamics at Homoclinic Bifurcations*, Cambridge Univ. Press, Cambridge, UK, 1993.

[18] T. S. Parker and L. O. Chua, *Practical Numerical Algorithms for Chaotic Systems*, Springer, Berlin, 1989.

[19] C. Simó, *On the analytical and numerical approximation of invariant manifolds*, in Les Méthodes Modernes de la Mécanique Céleste, edited by D. Benest and C. Froeschlé, Goutelas, (1989) pp. 285.

[20] S. H. Strogatz *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering* Reading, MA: Perseus Books, Cambridge MA, 1994.

[21] R. Szalai, *Nonlinear Vibrations of Interrupted Cutting Processes*, MSc thesis, Budapest University of Technology and Economics, 2002.

[22] D. Sweet, H. E. Nusse and J. A. Yorke, *Stagger-and-step method: detecting and computing chaotic saddles in higher dimensions*, Physical Review Letters, Vol. 86, No. 11 (2001) pp 2261–2264.

[23] Z. You, E. J. Kostelich and J. A. Yorke, *Calculating stable and unstable manifolds*, Int. J. Bifurcation and Chaos, Vol1, No. 3 (1991) pp. 605–623.

[24] J. P. England and H. M. Osinga *Global Manifolds 1D Version 2*, software for use with DsTool (2003), `http://www.enm.bris.ac.uk/staff/hinke/dss/map/GM1D/`.