An Algorithm for the Computation of Preimages in Noninvertible Mappings

Chia-Hsing Nien   Frederick J. Wicklin

The Geometry Center and School of Mathematics

University of Minnesota, Minneapolis, MN 55454

June 9, 1997

Shortened title: Computation of Preimages

Abstract

For discrete dynamical systems generated by iterating a diffeomorphism, every point in the phase space has a unique preimage and it is straightforward to compute geometric structures such as inverse orbits and one-dimensional stable manifolds of periodic points. For noninvertible mappings, however, some points have multiple preimages; others may have no preimages. This makes the computation of inverse orbits difficult, because accurate computations require global knowledge about the way the mapping folds and pleats phase space.

In this article we use ideas from singularity theory to examine the geometry of noninvertible mappings. We use the geometry to derive a computational algorithm for efficiently computing preimages in noninvertible mappings.

1 Introduction

For discrete dynamical systems of a single real variable, much of the geometry underlying the dynamics is understood by knowing the location of the critical points in the domain of a mapping, and then following the iterates of the critical points [Devaney, 1985]. The generalization of this concept to iterated mappings of two real variables involves locating points in the domain where the Jacobian of the mapping is singular. This typically occurs along curves in the domain [Mira & Narayaninsamy 1993].

Let $F\colon D \subseteq \mathbb{R}^2 \to \mathbb{R}^2$ be a smooth mapping (F may have poles outside of D). If F is not one-to-one, then the projection of the graph of F into the range of F will cover some regions of the range multiple times. This allows us to divide the range of F into open regions: regions not covered by the graph of F, regions covered exactly once, regions that are double-covered, and so forth.

Whitney [1955] showed that generic singularities of F are folds and pleats. If the graph of F has a fold in it, then locally the projection of the fold onto the range is a curve that separates a region with k preimages from regions with $k+2$ preimages. If two fold curves meet, they generically form a pleat, meaning that the projection of the fold curves meet at a cusp point. Although this curve is not smooth, it is typically the image of a smooth curve in the domain.

This article uses the geometry of pleats and folds to construct an algorithm to compute preimages of a point in the phase space of a noninvertible mapping. This enables researchers to examine the structure of the set of points that are mapped onto a given point by F^i for some finite i. Of course, if it is possible to explicitly compute each branch of the inverse for a noninvertible mapping (e.g., [Frouzakis *et al.*, 1992]), then this algorithm would not be necessary, but in general it is impossible to analytically invert a noninvertible map, and so a numerical approximation must be constructed.

The singularity theory used in Sec. 3 of this article is not new, but we believe it is important to spell out how these ideas apply to noninvertible dynamical systems. For example, this language may be applied to the work of Mira *et al.* [1995], which classifies noninvertible mappings in terms of regions in the range of a mapping which have a distinct number of preimages. In fact, the subject of this article was inspired by [Mira *et al.*, 1995], and by presentations by Mira and his colleagues at a conference on Noninvertible Dynamical Systems held at the Geometry Center, March 1995.

2 A Basic Algorithm to Compute Preimages

In order to make the presentation of the algorithm as clear as possible, we first present the algorithm in the context of one dimensional noninvertible mappings. The geometry in this situation is very easy to describe and visualize, and the main features of the algorithm are readily apparent.

2.1 A One-Dimensional Example

Let's begin with a very concrete example: a unimodal mapping of the interval such as the logistic map $f_\mu(x) = \mu x(1-x)$, $x \in [0,1]$, $\mu < 4$. Suppose that we want to compute preimages of an arbitrary point in the interval, and let's ignore the fact that we can explicitly compute the two inverse branches for this mapping.

For any mapping F, it is useful to define the critical curve

$$J_0 = \{x \mid DF(x) \text{ is singular}\},$$

which in the one-dimensional case reduces to the set of critical points for which the derivative vanishes. It is also useful to define the image of this set:

$$J_1 = F(J_0).$$

The set J_1 divides range into regions with constant number of preimages. For planar mappings, the sets J_0 and J_1 are also known as LC_{-1} and LC, respectively [Frouzakis *et al.*, 1997].

For the logistic map, J_1 is a single point that divides the interval into two intervals. Points in the interval $[0, f_\mu(1/2))$ are double-covered by the projection of the graph of f_μ, and so have two preimages, whereas points in the interval $(f_\mu(1/2), 1]$ have no preimages.

The graph of a less trivial one-dimensional mapping, F, is shown in Fig. 1. Here J_0 consists of three points, and J_1 divides the interval into regions with zero, 2, or 4 preimages. The function F is a seventh degree polynomial defined by

$$F(x) = \frac{3}{2}\left(\frac{5}{8} + \left(\frac{x}{2} - \frac{3}{4}\right)^3\right) f^2_{10/3}(x)$$

where $f^2_{10/3}$ is the second iterate of the logistic function with parameter $\mu = 10/3$.

The simple observation that underlies the preimage algorithm is that *our mapping is a diffeomorphism in regions that do not intersect J_0. So if we triangulate the domain of our mapping so that J_0 is on the boundary of the subdivision, then our mapping is a diffeomorphism on the interior of each cell of the triangulation.*

Let's see how we can exploit this fact. In one dimension, a triangulation is just a subdivision, so suppose we partition the domain shown in Fig. 1 into five subintervals ([0,1/5], [1/5,2/5], [2/5, 3/5], [3/5,4/5], [4/5,1]). We then refine the triangulation so that J_0 is on the boundary of the subdivision.

Now approximate F by the piecewise-linear mapping, $\hat{F}$, that agrees with F on each vertex of the triangulation, and is linear on the interior of each cell.

Enumerate each cell of the triangulation, and for each cell C, compute which other cells intersect $\hat{F}(C)$. The collected data is presented in Table 1 for this one-dimensional example.

Preprocessing Step		
Cell	Interval	$\hat{F}(C_i)$ Intersects
C_0	$[0, 0.2]$	C_0, C_1, C_2, C_3
C_1	$[0.2, a]$	C_2, C_3
C_2	$[a, 0.4]$	C_2, C_3
C_3	$[0.4, b]$	C_2
C_4	$[b, 0.6]$	C_2, C_3
C_5	$[0.6, 0.8]$	C_3, C_4, C_5
C_6	$[0.8, c]$	C_5
C_7	$[c, 1]$	$C_0, C_1, C_2, C_3, C_4, C_5$

Table 1: Preprocessing step for a one-dimensional example. The critical points of the function within $[0, 1]$ are located at $a \approx 0.240635, b \approx 0.474424, c \approx 0.824135$.

Once this data is obtained, we may compute preimages of points very rapidly. Given any point y, locate the cell C that contains y. If C_i, $i = 1 \dots k$ are the cells whose image under $\hat{F}$ intersect C, then use this intersection data and simple linear algebra to "pull back" y under $\hat{F}|_{C_i}$. Each of these linear pullbacks, t_i, may then be used as a seed for a root-finding algorithm such as Newton's method, in order to locate the true point preimage x_i near t_i that is mapped by F onto y.

For example, suppose we want to compute the preimage of the fixed point $y = 0.37926$. We see that this point is contained in the cell C_2. From the

Table, we see that y is potentially contained in $\hat{F}(C_i)$ for $i = 0, 1, 2, 3, 4, 7$. A quick check of the actual coordinates of the $\hat{F}(C_i)$ reveals that in fact y is only contained in the images for $i = 0, 2, 4, 7$. The linear function $\hat{F}|_{C_i}$ is easily inverted for each i to find the linear preimages of y in C_0, C_2, C_4, and C_7. A few iterations of Newton's method quickly locates the preimages of y under F.

2.2 A Two-Dimensional Extension

In this subsection we extend this algorithm to a planar mapping, F. While the ideas of the algorithm apply to maps on R^n for arbitrary n, implementing the details becomes increasingly difficult and the algorithm consumes a great deal of computer memory for $n \geq 3$.

The main difficulty encountered in the two-dimensional case is that of computing the curve J_0, and then triangulating the domain so that J_0 is not interior to any cells of the triangulation. Of course, in general, this is an impossible requirement, since $J_0 = \{x \in \mathbb{R}^2 | \det DF(x) = 0\}$ will typically not be a piecewise-linear curve! Examine, however, the geometric reason for this requirement: we want the mapping to be a diffeomorphism on the interior of each cell, and to fold the phase space along the boundary of certain cells. Since J_0 will typically be computed by continuation methods in the phase space [Allgower & Georg, 1990], the curve will be known to us as a piecewise-linear curve, so we can use the computed representation of J_0 to refine our triangulation.

An even more efficient approach, however, is to compute a triangulation of the phase space while *simultaneously* finding J_0. We can accomplish this using a piecewise-linear continuation algorithm [Allgower & Georg, 1990, Chap. 12], preferably one that uses an adaptive mesh refinement. Regardless of the mesh used, each cell, C, in the triangulation should be small enough that $F|_C \approx \hat{F}|_C$, where $\hat{F}$ is the piecewise-linear function that agrees with F on the vertices of the triangulation, and is linear on the interior of each cell.

Once this step is completed, the remainder of the algorithm proceeds as in the one dimensional case. Note that it is still possible to break up the algorithm into two parts: a *preprocessing step* that only needs to be done once for each mapping, and a *preimage step* that can be done many times very quickly, once the preprocessing is complete. The complete algorithm for computing preimages of F is therefore:

Preprocessing Step (Done once)

- **Compute** $J_0 = \{x | \det DF(x) = 0\}$
- **Triangulate** phase space so that J_0 is approximated by edges of the triangulation.
- **For each cell** C, record all cells, C_i, such that $\hat{F}(C_i) \cap C \neq \emptyset$.

For each point y that we want preimages of

- **Locate** the cell C that contains y
- **For each cell** C_i computed in the preprocessing step, compute the linear preimage $x_i = \left(\hat{F}|_{C_i}\right)^{-1}(y)$
- **Use a Root-finding Algorithm** to iterate each linear pullback x_i onto the real preimage.

2.3 Modifications to the Basic Algorithm

There are a few simple modification to the basic algorithm that can improve the performance of the algorithm.

As mentioned previously, the simplest modification is to use an adaptive piecewise-linear continuation algorithm to simultaneously compute J_0 and a triangulation of phase space. We implemented such an algorithm in Pisces [Wicklin, 1995], and, in fact, implemented the entire preimage algorithm in that environment. An example of an adaptive grid decomposition of a region of the plane is shown in Fig. 2 for the mapping

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

We remark that the last step of the preprocessing must be carried out efficiently, or it may dominate the computational time required by the algorithm. Clearly, it is not desirable to search each cell to see if its image intersects a given cell. One way to efficiently implement this step is to associate to the triangulation a quad-tree data structure that allows one to easily look-up the indices of cells whose images are located in a certain region of phase space.

While the use of an iterative root-finding algorithm makes it virtually impossible to prove, for example, that the algorithm computes all preimages correctly for all maps, our experience has been that the algorithm performs extremely well on a diverse set of mappings: polynomial, rational, and transcendental. The underlying mathematical idea behind the algorithm is that of simplicial approximation (e.g., [Henle, 1979]): as the triangulation becomes finer, the piecewise-linear mapping $\hat{F}$ converges to F.

It is easy to construct examples for which the algorithm will fail to detect all preimages, but the most typical instance in which this occurs is when attempting to compute one or more preimages that are near J_0. Choosing an adaptive triangulation that is more refined near J_0 helps reduce this difficulty, but since most root-finding algorithms fail near J_0, we do not know how to completely eliminate the difficulty of computing preimages that lie near J_0.

2.4 A Two-dimensional Example

To illustrate the algorithm, we compute preimages of a fixed point of Eqn. (1). We choose the saddle point located at $(0, -b/a)$. The J_0 curve for F is the graph of a cubic curve. The corresponding J_1 curve is a swallowtail; two portions of the swallowtail intersect the bounding box of the figure, including the lower cusp of the "tail."

This saddle point has four preimages, as do all point inside the swallowtail. Note that points immediately outside of the swallowtail have two regions, whereas points in the upper left-hand corner of the Figure have zero preimages.

Figure 3 shows points that iterate onto the saddle point within 18 iterates. The points that are iterated onto the saddle point by F^i, $i = 0, 1, 2, 3$ are labeled. The remaining dots indicate points on the stable manifold (or, to be more precise, the stable *set* [Frouzakis *et al.*, 1997]), but have the additional special property that they are eventually fixed. The two points indicated by boxes are a period-two source orbit; the twelve large crosses are a period-twelve orbit of saddle type. These periodic points are included to help the reader understand why certain portions of the Figure are "clumpy" or sparse; obviously, these orbits are not preimages of the fixed point.

Note for this mapping that the line $x = 0$ is the unstable manifold: the line is forward invariant and the map restricted to this line is a linear expansion for $a > 0$. The Figure indicates that some preimages of the fixed point accumulate on the unstable manifold; others accumulate on preimages of the unstable manifold. Preimages that are close to the local unstable manifold of the saddle point will themselves have four preimages that will be located near the points labeled with 0 and 1 in the Figure.

It is not the intent of this paper to give a complete analysis of the dynamics of this mapping, but rather to indicate the type of insight that is obtained by computing preimages. For example, the computation of preimages suggests that there are homoclinic trajectories for this mapping. It also suggests that symbolic dynamics would be useful for explaining the preimage set, since the set seems to decompose into teardrop-shaped clusters that are mapped onto each other by F.

The computation also suggests further questions and explorations. For example, how does the set of preimages bifurcate as parameters change? The most obvious change for this mapping occurs as the parameter b is varied and the fixed point moves outside of the swallowtail region, but there are other bifurcations, such as when the swallowtail vanishes, or when J_0 crosses the preimage set, resulting in some points having no further preimages.

3 Degenerate points of J_0

3.1 Cusp points

The occurrence of cusp points on J_1 is a generic feature of planar mappings. The implication of this fact for the preimage algorithm is important, and is illustrated in Fig. 4. As Fig. 4.A shows, the basic algorithm will mistakenly predict that points "inside" and near the cusp will have two fewer preimages than they actually have. A solution to this problem is to detect the preimages of cusp points (which we will call a *precusp*, although Golubitsky and Guillemin merely call it a "cusp point" [Golubitsky & Guillemin, 1973]), and to locally refine the triangulation so as to use the precusp as a vertex in the triangulation. This is illustrated in Fig. 4.B.

To compute the location of a precusp, we apply a little singularity theory (cf. [Arnold, 1981] and [Gilmore, 1981]). By definition, DF is singular for points on the critical curve, J_0. Generically, the kernel of DF will be one-dimensional and will vary continuously along the critical curve. Similarly, the tangent to the critical curve will vary continuously along the curve. It is not surprising, then that sometimes the tangent space to the critical curve at p will be contained in the kernel of $DF(p)$.

Since F maps J_0 to J_1, DF maps tangent vectors at a point on J_0 to tangent vectors at the image point on J_1. Typically, non-zero tangent vectors in the domain of DF are sent to non-zero vectors in the range. If p is a precusp point, however, then a non-zero tangent vector at p is sent to the zero vector at $F(p)$, meaning that the J_1 curve does not have a well-defined tangent space at $F(p)$. This is the defining characteristic for the existence of a cusp.

Let $G = DF \cdot T$ be defined on J_0, where T is a unit tangent vector to J_0. Algebraically, the condition that the kernel contains the tangent space is just that $G = 0$, or that T is an eigenvector of DF with eigenvalue 0. We remark that the condition that T be a unit vector can be relaxed. In fact, for planar maps we will often use $(-f_y, f_x)$ as the tangent vector where $f = \det DF$.

Thus we can locate precusp points by finding locations where $G = 0$. Strictly speaking, we should not label a point as "precusp" until we have verified additional nondegenerate conditions related to the nonsingular nature of DG at the point in question, but we omit this computation.

3.2 Singular Curves: Beak-to-beak and Lips

It is often the case that we want to compute stable sets for a range of parameter values and to study the bifurcations that occur as the parameters vary. Morse theory [Golubitsky & Guillemin, 1973, Chap. II.6] tells us that level sets of a continuous function (in this case, $f = \det DF$) can only locally change their topology by passing through a singularity. If p is a point on a critical curve J_0, p is a *singular point* for J_0 if $\nabla f(p) = 0$. If $\det D^2 f(p) \neq 0$, then p is a *nondegenerate* singular point. If $\det D^2 f(p) < 0$, we say that p is of *nodal type*; if $\det D^2 f(p) > 0$, we say that p is of *isola type*.

The typical transitions of J_0 and J_1 as J_0 passes through each kind of singularity are shown in Figs. 5 and 6. Because of the characteristic shape of J_1, the transition through the singularity of isola type is called the *lips metamorphosis*, whereas the transition through the singularity of nodal type is called the *beak-to-beak metamorphosis* [Gilmore, 1981]. As was the case for the cusp singularity, we should ensure that the singular point of J_0 becomes a vertex in our refined triangulation. Furthermore, the triangulation should be locally refined in the neighborhood of the singularity.

4 Acknowledgements

The authors thank J. Guckenheimer, R. McGehee, R. Moeckel, and J. Yorke for helpful discussions related to this work. All of the critical curves and preimages in this article were computed by `Pisces` [Wicklin, 1995]. The Pisces Project was supported by the NSF-sponsored Geometry Center.

References

[1] Allgower, E. L. & Georg, K. [1990] *Numerical continuation methods, an introduction*, Springer-Verlag.

[2] Arnold, V. I. [1981] *Singularity theory*, London Mathematical Society.

[3] Devaney, R. [1985] *An introduction to chaotic dynamical systems*, Benjamin Cummings, Menlo Park, CA.

[4] Frouzakis, C. E., Adomaitis, R. A., Kevrekidis, I. G., Golden, M. P. & Ydstie, B. E. [1992] *The structure of basin boundaries in a simple adaptive*

control system, Chaotic Dynamics: Theory and Practice (Bountis, T. ed.), Plenum Press, pp. 195–210.

[5] Frouzakis, C. E., Gardini, L., Kevrekidis, I. G., Millerioux, G. & Mira, C. [1997] *On some properties of invariant sets of two-dimensional noninvertible maps*, to appear, Int. J. Bifur & Chaos.

[6] Gilmore, R. [1981]*Catastrophe theory for scientists and engineers*, Wiley.

[7] Golubitsky, M. & Guillemin, V. [1973] *Stable mappings and their singularities*, Springer-Verlag, New York.

[8] Henle, M. [1979] *A combinatorial introduction to topology*, Dover.

[9] Mira, C., Carcassès, J-P., Millérioux, G. & Gardini, L. [1995]*Plane foliations of two-dimensional non invertible maps*, Tech. report, Groupe d'Etudes des Systémes Non Linèaires et Applications, LESIA.

[10] Mira, C. & Narayaninsamy, T. [1993] *On behaviors of two-dimensional endomorphisms: Role of the critical curves*, Int. J. Bif. Chaos **3**, no. 1, 187–194.

[11] Whitney, H. [1955] *On singularities of mappings of Euclidean space I: Mappings of the plane into the plane*, Annals of Mathematics **62**, 374–410.

[12] Wicklin, F. J. [1995] *Pisces: A platform for implicit surfaces and curves and the exploration of singularities*, Tech. Report GCG #89, The Geometry Center.

Figure 1: A one-dimensional mapping with multiple critical points, illustrating how J_1 separates the domain into regions of a constant number of preimages.

Figure 2: An adaptive grid decomposition of a region for the planar quartic mapping defined in Eqn. (1). Here $\mu = a = b = \epsilon = 0.1$ and $\nu = -1.9$.

Figure 3: A computation of points mapped onto the saddle point at $(0, -b/a)$ within 18 iterations. The parameters are the same as for the previous Figure.

Figure 4: (A) Two triangles, one of which contains a precusp point, and the image of those triangles under $\hat{F}$. The algorithm will be unable to find preimages of points in the shaded region. (B) After detecting the precusp point and locally refining the triangulation by adding a new vertex at the precusp point, the triangulation can better represent the folding and pleating of phase space.

Figure 5: The critical curves J_0 for different parameter pairs (μ, ν) in the family $(x, y) \mapsto (\frac{1}{3}x^3 + \mu x + \nu x y^2, y)$. (A): $(-1, 1)$; (B): $(0, 1)$; (C): $(1, 1)$; (D): $(-1, -1)$; (E): $(0, -1)$; (F): $(1, -1)$. The curves are all plotted in the region $[-2, 2] \times [-2, 2]$.

Figure 6: The corresponding J_1 curves of Figure 5. The curves are plotted in the region $[-2, 2] \times [-2, 2]$.