Computing specific isolating neighborhoods

Oliver Junge
Department of Mathematics and Computer Science
University of Paderborn
33098 Paderborn, Germany
junge@upb.de

October 30, 2001

Abstract

A major challenge in applying the Conley index theory to a given dynamical system is to find *isolating neighborhoods* of the invariant sets of interest. In this paper we describe a computational method which allows for the efficient construction of isolating neighborhoods. As an example we compute an isolating neighborhood of a heteroclinic orbit connecting a fixed point to a period two point in the Hénon map.

Key words. Conley index, isolating neighborhood, invariant set.

Introduction

In [5] Szymczak presents a method to construct isolating neighborhoods and index pairs for a given discrete dynamical system. This is one of the main computational challenges in applying the Conley index theory to a concrete dynamical system. Szymczak's approach is to represent the sets of interest as unions of *cubes* (*boxes*) and to design a combinatorial algorithm constructing a cubical set which is an isolating neighborhood for the given system. Once such a neighborhood has been found the construction of a corresponding index pair is immediate. The algorithm in [5] successively removes boxes from an inital (guessed) collection of boxes until the remaining ones satisfy some condition which garantuees that this cubical set actually is an isolating neighborhood. The central problem of this approach is to find a good initial guess, one that after termination of the algorithm leads to some "interesting" isolating neighborhood (i.e. a corresponding index computation does reveal some interesting facts about the dynamics).

In this paper we advocate a different approach for assembling the desired cubical set, which has been used in [1] for the analysis of an infinite dimensional map. Rather than starting from some big collection and successively removing boxes from it, we start with a guess for some specific invariant set and successively *add* boxes to that collection until the criterion for isolation is fulfilled. Certain initial guesses are easily obtained by considering the *transition matrix* P of the system – periodic points correspond to nonzero diagonal entries of some power of P, connecting orbits are approximately located by computing the *shortest path* between two (periodic) cubes.

As an example we construct a neighborhood which isolates a heteroclinic orbit connecting a fixed point to a period two point in the Hénon map.

Isolating neighborhoods

We consider the temporal evolution of a system given by a smooth map $f : X \to X$, where X is some compact subset of $\mathbb{R}^n$. A (full) trajectory of f is a sequence $\sigma_x : \mathbb{Z} \to X$ satisfying $\sigma_x(0) = x$ and $\sigma_x(i+1) = f(\sigma_x(i))$ for all $i \in \mathbb{Z}$. A set $S \subset X$ is called *invariant*, if for every $x \in S$ there is a full solution $\sigma_x : \mathbb{Z} \to S$. A compact set $I \subset X$ is an *isolating neighborhood* if its maximal invariant set is contained in its interior, i.e. if

$$\mathrm{Inv}(I, f) = \{x \in I \mid \exists\, \sigma_x : \mathbb{Z} \to I\} \subset \mathrm{int}(I).$$

Recall that our aim is to computationally construct isolating neighborhoods for specific invariant sets of f. The first question is how to represent these sets in the computer. To this end we are going to discretize the phase space X into a finite number of cubical sets. A *cubical set* is a subset B of $\mathbb{R}^n$ of the form

$$B = B(c, r) = \{x \mid |x_k - c_k| \leq r_k, k = 1, \ldots, n\},$$

where $c, r \in \mathbb{R}^n$ and $r_k \geq 0$. From now on we assume that X is a cubical set. Let $c = (c_1, \ldots, c_n)$ be the center and $r = (r_1, \ldots, r_n)$ be the radius of X, then by *bisecting* X with respect to the j-th coordinate direction one obtains two cubical sets $B(c^-, \hat{r})$ and $B(c^+, \hat{r})$, where

$$\hat{r}_k = \begin{cases} r_k & \text{for } k \neq j \\ r_k/2 & \text{for } k = j \end{cases}, \quad c_k^{\pm} = \begin{cases} c_k & \text{for } k \neq j \\ c_k \pm r_k/2 & \text{for } k = j \end{cases}.$$

A cubical set which can be represented by iterating this subdivision process will be called a *box*. Note that a binary tree represents a certain set of boxes if one assigns a coordinate direction to each depth of the tree: the root corresponds to the box X and all nodes of a given depth correspond to a subset of a cubical grid on X (i.e. a partition of X into cubical sets) – see [3] for a more detailed description of this hierachical way of storing cubical grids. Denote by $\mathcal{B}_k$ the collection of all boxes represented by the nodes of depth k of some tree (where the root has depth 0). For a subset $\mathcal{B} \subset \mathcal{B}_k$ let $|\mathcal{B}|$ denote the union of all boxes in $\mathcal{B}$. Let $o(\mathcal{B})$ be the set of all boxes in $\mathcal{B}_k$ which intersect $|\mathcal{B}|$, i.e. the smallest representable neighborhood of $|\mathcal{B}|$ in $\mathcal{B}_k$.

The map f defines in a natural way a multivalued map $\mathcal{F}$ on $\mathcal{B}_k$: For $B \in \mathcal{B}_k$ let $\mathcal{F}(B)$ be the set of all boxes in $\mathcal{B}_k$ which intersect the set $f(B)$. However, in order to allow for errors introduced when computing and representing $f(B)$ and since the following object will be simpler to compute, we will actually deal with some *enclosure* of f, i.e. a map $\mathcal{F} : \mathcal{B}_k \rightrightarrows \mathcal{B}_k$ such that

$$f(B) \subset \text{int}|\mathcal{F}(B)| \tag{1}$$

for $B \in \mathcal{B}_k$. It is the map $\mathcal{F}$ that we are going to deal with in the computer. In an implementation it may be represented as e.g. a (sparse) matrix or – equivalently – a graph.

Computing an enclosure of f. An approximate method to compute all boxes B' in the collection $\mathcal{B}_k$ which intersect the image $f(B)$ of a given box $B \in \mathcal{B}_k$ is to choose a finite set T of *test points* in B and to consider all boxes B' which contain at least one of the image points $f(T)$. This is the approach originally proposed in [3]. Note that due to the hierachical storage scheme of the boxes in a binary tree the computational complexity of determining the box $B' \in \mathcal{B}_k$ which contains some specific image point is only $\mathcal{O}(k)$. In general this approach will not yield an enclosure of f. However, in [4] it has been shown how to extend this approach to compute an enclosure by constructing an appropriate mesh of test points in each box.

The approach used here is still different (and more efficient) and based on the following observations:

1. For small enough boxes (i.e. large k) the image of a box $B \in \mathcal{B}_k$ under the map f is approximately given by its image under the linear part of f;

2. For a given cubical set $C \subset \mathbb{R}^n$ the set of all boxes $B' \in \mathcal{B}_k$ which intersect C can efficiently be determined by a single depth first search in the tree;

So the idea is to use the linear part of f to compute an approximate image of a box B, to enclose this image by a cubical set and then to enlarge this cubical set by the errors made by neglecting the nonlinear terms of f. In doing these computations we use interval arithmetic in order to control round-off errors.

Let us be more precise. Consider the box $B = B(c, r) \in \mathcal{B}_k$. For $h \in \mathbb{R}^n$ we can decompose f as

$$f(c + h) = f(c) + Df(c)h + f^{nl}(c, h),$$

where $f^{nl}(c, 0) = 0$. Let

$$\varepsilon_k^{nl}(c) = \max_{\substack{h:|h_j|\le r_j \\ j=1,\dots,n}} |f_k^{nl}(c, h)|, \quad k = 1, \dots, n,$$

then $f(B)$ will be contained in the cubical set $B(c, R)$,

$$R = R(c, r) = |Df(c)|r + \varepsilon^{nl}(c),$$

where for a matrix $A = (a_{ij}) \in \mathbb{R}^{d,d}$ we write $|A| := (|a_{ij}|) \in \mathbb{R}^{d,d}$. One should emphasize that the computation of ε^{nl} may in general be expensive – it is not in our case. Now the enclosure $\mathcal{F} : \mathcal{B}_k \rightrightarrows \mathcal{B}_k$ of f is defined in the following way: Let $\mathcal{F}(B(c, r))$ be the set of boxes which is intersected by the cubical set $B(f(c), R(c, r))$.

The following algorithm (when called as $\texttt{cap}(X, C, k)$) computes the set $\mathcal{I}$ of all boxes in $\mathcal{B}_k$ which have a nonempty intersection with the cubical set C. Its complexity is bounded by $\mathcal{O}(kn)$, where n is the cardinality of $\mathcal{I}$.

Algorithm 1.

$\mathcal{I}$ = $\texttt{cap}(B, C, k)$
    if $B \cap C \neq \emptyset$
        if $\texttt{depth}(B) = k$
            $\mathcal{I} := \mathcal{I} \cup \{B\}$
        else
            $\mathcal{I} := \mathcal{I} \cup \texttt{cap}(B^+, C, k) \cup \texttt{cap}(B^-, C, k)$
    return $\mathcal{I}$

Here the function $\texttt{depth}(B)$ returns k if $B \in \mathcal{B}_k$, and B^+ and B^- are those two boxes which result from bisecting B with respect to some coordinate direction.

So far we constructed a discretized version of f on a cubical grid on X. Let us discretize the notion of an isolating neighborhood, too, now. A (full) trajectory of $\mathcal{F} : \mathcal{B}_k \rightrightarrows \mathcal{B}_k$ is a sequence $\sigma_B : \mathbb{Z} \to \mathcal{B}_k$ satisfying $\sigma_B(0) = B$

and $\sigma_B(k+1) \in \mathcal{F}(\sigma_B(k))$ for all $k \in \mathbb{Z}$. The *maximal invariant set* of $\mathcal{F}$ in a subset $\mathcal{B}$ of $\mathcal{B}_k$ is

$$\mathrm{Inv}(\mathcal{B}, \mathcal{F}) := \{B \in \mathcal{B} \mid \text{there exists a full trajectory } \sigma_B : \mathbb{Z} \to \mathcal{B}\}.$$

$\mathcal{B}$ is an *isolating neighborhood* if

$$o(\mathrm{Inv}(\mathcal{B})) \subset \mathcal{B}. \tag{2}$$

An important result following from (1) is that if $\mathcal{B}$ is an isolating neighborhood for $\mathcal{F}$, then $|\mathcal{B}|$ is an isolating neighborhood for f, see [5] for a proof.

Computing isolating neighborhoods. Szymczak ([5]) describes an algorithm for finding isolating neighborhoods of f in a given subset $\mathcal{B}$ of $\mathcal{B}_k$. The basic idea is to "cut" the right pieces out of $\mathcal{B}$ until the resulting collection satisfies (2). This method has the drawback that one has to choose a suitable set $\mathcal{B}$ apriori and may eventually end up with the empty set as a trivial isolating neighborhood. The approach we will describe now proceeds in some sense in the opposite direction: one starts with a *guess* for an isolating neighborhood of some interesting invariant set and "fattens" this set by adding neighborhoods until the condition (2) is satisfied.

Guesses for isolating neighborhoods of specific invariant sets of $\mathcal{F}$ can easily be obtained:

- k-periodic points of $\mathcal{F}$ are identified by nonzero diagonal entries of $M_{\mathcal{F}}^k$, where the *transition matrix* $M_{\mathcal{F}} = (m_{ij})$ is given by

$$m_{ij} = \begin{cases} 1, & \text{if } B_j \in \mathcal{F}(B_i) \\ 0, & \text{else} \end{cases}$$

  and $\mathcal{B}_k = \{B_1, \dots, B_p\}$;

- more generally, recurrent sets of $\mathcal{F}$ are given by strongly connected components of a graph representing $\mathcal{F}$;

- in that graph connecting orbits of $\mathcal{F}$ can be identified by shortest path algorithms (i.e. the Dijkstra-algorithm);

- finally Szymzcak [5] describes how to compute the maximal invariant set of $\mathcal{F}$.

Note that one easily obtains guesses for invariant sets with complicated dynamics by constructing a guess for a *heteroclinic cycle* between two periodic points – see [1] for details and examples.

Once a guess $\tilde{\mathcal{I}}$ for an isolating neighborhood of $\mathcal{F}$ has been computed, we construct a (true) isolating neighborhood $\mathcal{I}$ containing $\tilde{\mathcal{I}}$ by the following procedure.

Algorithm 2.

```
𝓘 = make_isolated(𝓘̃)
    𝓘 := Inv(𝓘̃, 𝓕)
    while o(𝓘) ⊄ 𝓘̃
        𝓘̃ := 𝓘̃ ∪ o(𝓘)
        𝓘 := Inv(𝓘̃, 𝓕)
    if |𝓘| ⊂ int|o(𝓘)| return 𝓘
    else return ∅
```

By construction this algorithm returns an isolating neighborhood $\mathcal{I}$ for $\mathcal{F}$. Similar to the procedure proposed in [5] one may end up with the empty set, in which case the set $|\mathcal{I}|$ touched the boundary of X.

Example

As an example computation we construct an isolating neighborhood for a heteroclinic orbit between a fixed point and a period two point of the Hénon map. Note that this computation does not involve any apriori knowledge of the system's dynamics nor of the location of the computed objects. All the data structures and algorithms have been implemented within the software package GAIO[1].

We consider the Hénon map

$$f(x, y) = (1 - ax^2 + by, x)$$

with parameters $a = 1.4, b = 0.3$. We choose $X = [-1.5, 1.5]^2$ and construct a box-covering on depth 20 of the chain recurrent set of f within X (see [2] for a detailed description on how to accomplish this). We compute $\mathcal{F}$ on that collection and identify fixed and period two boxes via the transition matrix $\mathcal{M}_{\mathcal{F}}$. Finally, using a graph representation of $\mathcal{F}$, we compute the shortest path between a fixed box and a period two box and use the resulting box-collection as our guess for the isolating neighborhood. Application of algorithm 2 yields the box-collection as shown in Figure 1. The computation of a corresponding index pair and its Conley index proves that this cubical set indeed contains the desired objects (see e.g. [5, 1] for details on how to perform these two steps).

[1] http://www.upb.de/math/~agdellnitz/gaio

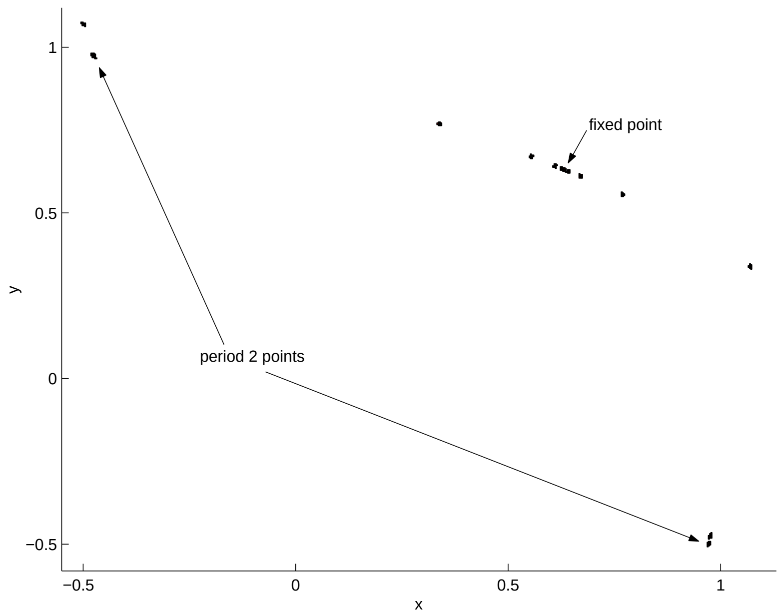

Figure 1: Isolating neighborhood for a heteroclinic orbit connecting a fixed point to a period two point in the Hénon map.

References

[1] M. Allili, S. Day, O. Junge, and K. Mischaikow. A rigorous numerical method for the global analysis of an infinite dimensional discrete dynamical system. Preprint, 2001.

[2] M. Dellnitz, O. Junge, M. Rumpf, and R. Strzodka. The computation of an unstable invariant set inside a cylinder containing a knotted flow. In *Proceedings of Equadiff '99, Berlin*, 2000.

[3] Michael Dellnitz and Andreas Hohmann. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numer. Math.*, 75(3):293–317, 1997.

[4] O. Junge. Rigorous discretization of subdivision techniques. In *Proceedings of Equadiff '99, Berlin*, 2000.

[5] Andrzej Szymczak. A combinatorial procedure for finding isolating neighbourhoods and index pairs. *Proc. Roy. Soc. Edinburgh Sect. A*, 127(5):1075–1088, 1997.