Computation of hyperbolic trajectories and their stable and unstable manifolds for oceanographic flows represented as data sets

A. M. Mancho, D. Small, and S. Wiggins

School of Mathematics, University Walk, University of Bristol, Bristol BS8 1 TW, UK

Part of Special Issue "Dedicated to Prof. A. D. Kirwan Jr. on the occasion of his 70th birthday"

Abstract. In this paper, we describe a unified algorithm for locating and computing hyperbolic trajectories, and then computing their stable and unstable manifolds, for finite time velocity fields in the form of a data set defined on a space-time grid. The algorithm is applied to a turbulent regime of a quasigeostrophic wind-driven double gyre in a rectangular domain.

1 Introduction

The applications of the dynamical systems approach to the study of Lagrangian transport in fluid mechanics is now well established with literally hundreds of papers related to the subject being published over the last twenty years, see the monographs Ottino (1989) and Wiggins (1992) for an introduction to the subject.

In recent years this approach has been applied to the study of Lagrangian transport in oceanographic flows, see, e.g. Behringer et al. (1991), Bower (1991), Samelson (1992), Dutkiewicz et al. (1993), Meyers (1994), Duan and Wiggins (1996), Rogerson et al. (1999), Poje and Haller (1999), Coulliette and Wiggins (2001), Yuan et al. (2001), Balasuriya and Jones (2001), Balasuriya (2001), Poje et al. (2002), Waseda and Mitsudera (2002), Kuznetsov et al. (2002), Miller et al. (2002), and Kirwan et al. (2003).

Hyperbolic fluid particle trajectories and their stable and unstable manifolds are the basic building blocks of the flow field on which the dynamical systems approach to Lagrangian transport is based. However, realistic oceanographic flows give rise to a number of mathematical obstacles to the implementation of these ideas. Such flows are typically aperiodic in time (so the usual idea of "Poincaré map" from dynamical systems theory is not applicable) and dynamically consistent flows can generally be realized only on a space-time grid, i.e. they are realized as a data set that is ob-

tained through the numerical simulation of a system of partial differential equations (as in this paper) or experimentally measured. In recent years some of the difficulties arising have been addressed by various groups; see, e.g. Duan and Wiggins (1997), Miller et al. (1997), Malhotra and Wiggins (1998), Haller and Poje (1998), Haller (2000), and Haller and Yuan (2000).

This paper is concerned with the computation of hyperbolic trajectories and their stable and unstable manifolds in oceanographic flows defined as data sets. We develop an improvement (to be described below) of algorithms given in Ide et al. (2002) and Ju et al. (2003) for computing hyperbolic trajectories. This algorithm is then combined with an algorithm given in Mancho et al. (2003) for computing the stable and unstable manifolds of hyperbolic trajectories. In previous work both of these algorithms have been implemented for simple, analytically know, velocity fields for the purpose of benchmarking. This paper is the first implementation of these algorithms for a turbulent velocity field defined as a finite time data set.

A hyperbolic fluid particle trajectory is the generalization to unsteady flows of a saddle-type stagnation point in a steady flow[1]. It could be thought of as a "moving saddle point". Its stable and unstable manifolds are the time-dependent generalizations of the separatrices of the saddle-type stagnation point of the steady flow. They are, likewise, material curves, which means that they cannot be crossed by other fluid particle trajectories. It is in this sense that they form the geometrical template for Lagrangian transport in flows.

Our search for hyperbolic trajectories is related to the structure of instantaneous stagnation points (ISP) in the frozen time flow. ISPs are related to the Eulerian flow structure, and themselves are not particle trajectories. Nevertheless, the work in Ide et al. (2002) and Ju et al. (2003) shows how hyperbolic ISPs can be used to initialize an iterative

---

Correspondence to: S. Wiggins
(s.wiggins@bristol.ac.uk)

[1]The definition of finite time hyperbolicity that we use in this paper is given in Appendix A.

process that may converge to a hyperbolic trajectory of the flow. This iterative method is based on certain integral equations and is therefore global in time for the entire length of the time interval of interest. This has the advantage of finding a hyperbolic trajectory that exists for the entire length of this time interval. Methods based on converging in time lose part of the data set in the process of convergence. For a finite time data set, this can be a severe limitation.

The work in Ide et al. (2002) applies only to flows where the region of the hyperbolic trajectory of interest can be described by a certain linearized velocity field (described more fully in Ide et al. (2002)). In this situation Fourier and convolution type methods can be used to construct a hyperbolic trajectory over the entire length of the finite time interval of interest. The algorithm developed in Ju et al. (2003) is fully nonlinear and, as mentioned above, utilizes an iterative technique to solve an integral equation, whose solution is a hyperbolic trajectory. The first step in this iterative procedure yields the answer one would obtain by the algorithm developed in Ide et al. (2002). The relation between the two algoritms is described in Ju et al. (2003). The algorithm for finding hyperbolic trajectories described in this paper is essentially that developed in Ju et al. (2003), but the implementation given here is slightly different and, we believe, more numerically advantageous.

Another important feature observed in complicated flows is that there can be many ISPs that appear and disappear in bifurcations at different times. Hyperbolic trajectories cannot bifurcate, since in order to bifurcate, hyperbolicity must be lost. The algorithms in Ide et al. (2002) and Ju et al. (2003) are not able to deal with this more complex case of bifurcating ISPs, which is essential in complex spatio-temporal flows. In this paper we provide a new extension of these algorithms to certain rather simple situations where this difficulty can be overcome (but the problem is far from being "solved", as we shall see).

Once a hyperbolic trajectory has been located we then want to compute its stable and unstable manifolds. Here we combine the hyperbolic trajectory algorithm described in this paper with the stable and unstable manifold computation algorithm described in Mancho et al. (2003) into a single, unified algorithm. As for most algorithms for computing stable and unstable manifolds of hyperbolic trajectories, this one involves evolving in time (forward in time for the unstable manifold, backward in time for the stable manifold) a small initial segment of the manifold centered on the hyperbolic trajectory. A significant feature of our approach is that this initial segment is provided by the hyperbolic trajectory locating algorithm. The evolution of the initial piece of the manifold is carried out adapting sophisticated numerical techniques developed by Dritschel (1989) and Dritschel and Ambaum (1997) for controlling the size of gaps between points in the manifold, interpolation, and point redistribution along the computed manifold. The numerical techniques of Dritschel and Ambaum were developed in the context of studies of the evolutions of the boundaries of vortex patches in complex fluid flows, and are ideal for the study of the evolution of stable and unstable manifolds in similar complex flows.

For a finite time data set one only needs to know a hyperbolic trajectory at the beginning and end of the time interval on which it is defined. For computing the unstable (resp. stable) manifold one goes to the beginning (resp. end) of the time interval and evolves forward (resp. backward) in time a small initial segment of the manifold centered on the hyperbolic trajectory. From this point in time onward one can "forget" about the hyperbolic trajectory since it is contained in the curve being evolved. Now it may be that for some known or unknown reasons the hyperbolic trajectory algorithm only converges on a time interval that is smaller than the length of time of the data set, yet in reality the hyperbolic trajectory exists outside this interval. This is simply a deficiency of the method (sometimes arising when we encounter bifurcations of ISPs). However, there is no reason the manifold (containing the hyperbolic trajectory) cannot be evolved in time outside the interval of time where the hyperbolic trajectory algorithm converged and the hyperbolic trajectory would be contained in the evolving curve. Of course, this brings up many new questions which are beyond the scope of this paper.

The unified algorithm for computing hyperbolic trajectories and their stable and unstable manifolds is then applied to a wind-driven quasigeostrophic double-gyre model. This is a continuation of related studies in Coulliette and Wiggins (2001), but at higher wind curl stress values, which yields a much more complex spatio-temporal flow.

2 The model: wind-driven double-gyre ocean circulation

Our algorithm will be applied to a model of mid-latitude wind-driven ocean circulation. The data set is obtained by a numerical simulation of a quasi-geostrophic (QG) 3-layer model in a rectangular basin geometry (Rowley, 1996). Due to the latitudinal antisymmetric wind stress curl applied at the ocean surface, the basic circulation pattern in the upper layer is a double-gyre structure separated by an eastward jet protruding into the flow from the confluence point of the southward and northward western boundary currents. Driven by the strong eastward jet, the northern and southern gyres circulate counterclockwise and clockwise, respectively. Depending on the ocean basin size and the amplitude of the wind stress curl, the ocean circulation exhibits a rich time-dependent dynamics (Dijkstra and Katsman, 1997) with realistic parameter values for the mid-latitude ocean (Lozier and Riser, 1989).

We demonstrate our algorithms for a wind stress curl at $0.32\,\mathrm{dyn/cm^2}$ where the ocean dynamics is turbulent. The velocity data set is obtained on a $1000\,\mathrm{km} \times 2000\,\mathrm{km}$ rectangular domain with spatial and temporal resolutions of $12.5\,\mathrm{km} \times 12.5\,\mathrm{km}$ and $2\,\mathrm{h}$ (although the data is only saved every $24\,\mathrm{h}$), respectively, and spans over 4000 days. The 4000 day interval of the data set is considered after the fluid
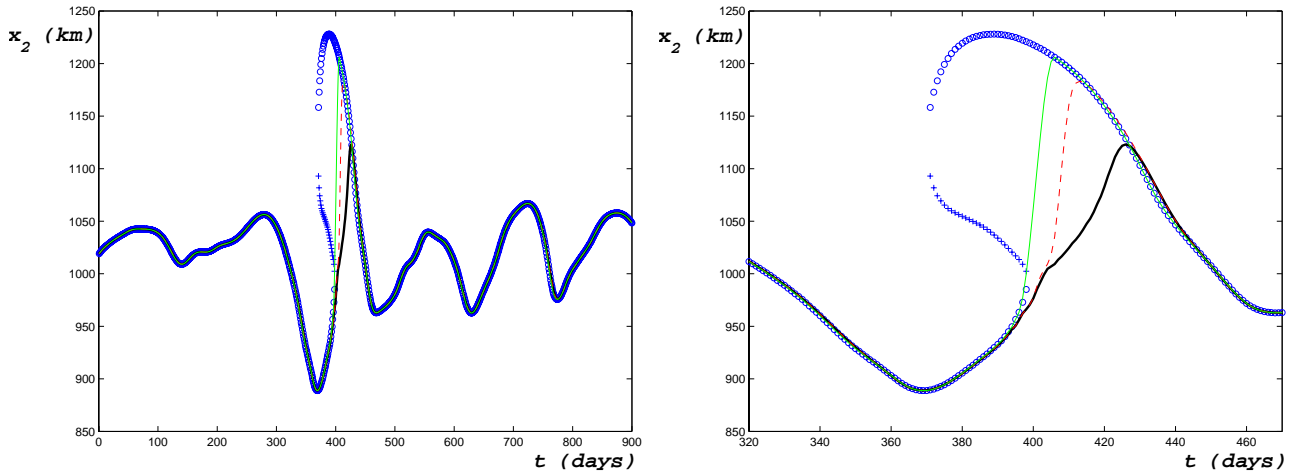
Fig. 1. Convergence to the hyperbolic trajectory on the western boundary. The right hand panel shows a blow-up of the region of bifurcation of the instantaneous stagnation points (blue). The path taken as the initial guess for the iterative procedure is show in green. The paths after the 10th iteration (dashed red line) and the 50th iteration (black), in which convergence is achieved, are also shown.

is started from rest and allowed to spin up for 25 000 days. In particular we have chosen the no-stress condition at the boundaries. Details of the numerical method used in solving the QG equations can be found in Rowley (1996) or Coulliette and Wiggins (2001). As the data is discrete in space and time, in order to have a continuous representation, necessary to integrate Eq. (1), we interpolate the velocity fields using bicubic interpolation in space and 3rd order Lagrange polynomials in time.

3 The numerical procedure for calculating hyperbolic trajectories and their stable and unstable manifolds from a velocity field given as a data set

In this section we describe the numerical approach to computing hyperbolic trajectories, and their stable and unstable manifolds, of two-dimensional velocity fields given as data sets.

We begin with a two-dimensional velocity field defined as a data set:

$$\dot{x} = v(x, t), \quad x = (x_1, x_2) \in \mathbb{R}^2, \quad t \in [t_0, t_L]. \tag{1}$$

First we compute a hyperbolic trajectory of Eq. (1).

3.1 Computation of hyperbolic trajectories

We begin with a descriptive overview of the method.

3.1.1 Overview of the Method for Computing Hyperbolic Trajectories

We utilize an iterative method, and like many iterative methods the success begins with a "good guess". This is a key, and somewhat subtle, point.

Our first guess is a path in space, denoted $x^{(0)}(t)$, $t \in [t_0, t_L]$, which has some form of hyperbolicity or "saddle-like" properties. It should be stressed that $x^{(0)}(t)$ is not a trajectory of Eq. (1) (or we would have "guessed the answer"). One possible guess is for $x^{(0)}(t)$ to be a curve of instantaneous stagnation points of Eq. (1), i.e. $v(x^{(0)}(t), t) = 0$, $t \in [t_0, t_L]$, and hyperbolicity is introduced by requiring that for each fixed t, $x^{(0)}(t)$ is a hyperbolic stagnation point of the frozen time velocity field. This is another subtle point. It is well established that hyperbolicity in the frozen time velocity field need not be reflected in hyperbolicity of the trajectories. So how is it useful for finding hyperbolic trajectories? We will clarify that point shortly. Hyperbolic frozen time stagnation points are a form of Eulerian hyperbolicity, from which we show how one may obtain Lagrangian hyperbolicity, i.e. hyperbolic fluid particle trajectories.

As one examines the stagnation points of the frozen time velocity field at different times, it may happen that hyperbolic stagnation points lose their hyperbolicity and disappear in bifurcations. In this case a hyperbolic stagnation point may not exist for the entire length (in time) of the data set. In this situation we will show some instances where the first guess may be a path in space where only part of the path is made up of hyperbolic stagnation points of the frozen time velocity field[2]. How to extend the path (in time) beyond the hyperbolic instantaneous stagnation point in this case requires some care. We will show some examples, but there is considerable room for further development here.

With the first guess, $x^{(0)}(t)$ in hand, we localize Eq. (1) around $x^{(0)}(t)$, i.e. letting

$$y = x - x^{(0)}(t), \tag{2}$$

---

[2]This typically occurs when two or more instantaneous stagnation points collide at some instant of time and disappear in a bifurcation. Hyperbolic trajectories cannot bifurcate in this manner.
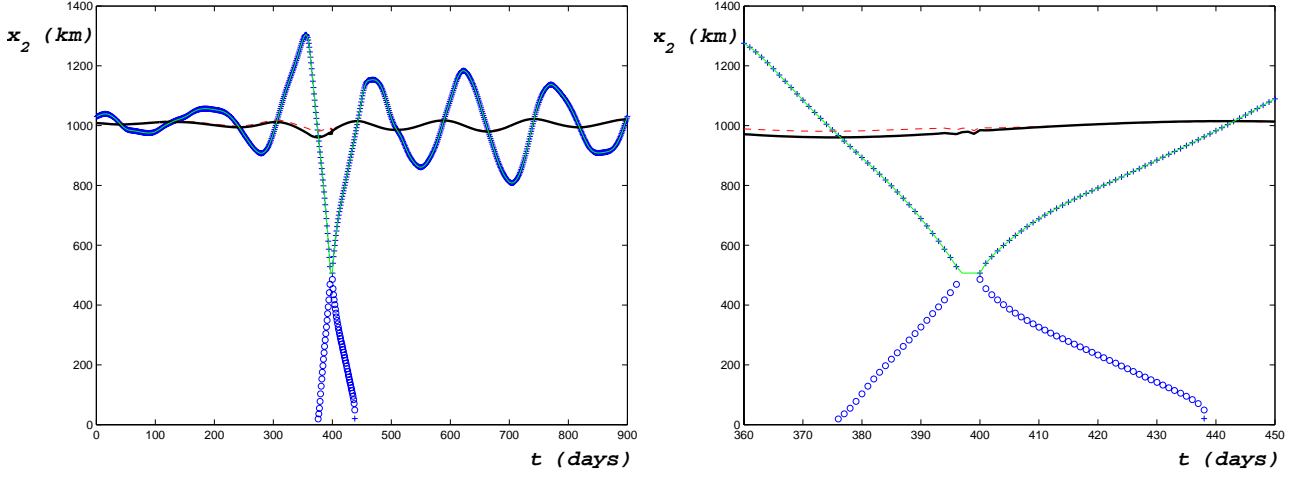
Fig. 2. Convergence to the hyperbolic trajectory on the eastern boundary. The right hand panel shows a blow-up of the region of bifurcation of the stagnation points (blue). The path taken as the initial guess for the iterative procedure is shown in green. The paths after the 1st iteration (dashed red line) and the 5th iteration (black), in which convergence is achieved, are also shown.

we rewrite Eq. (1) in the form:

$$y = \mathbf{A}^{(0)}(t) y + f^{(0)}(y, t), \tag{3}$$

where the precise form of $\mathbf{A}^{(0)}(t)$ and $f^{(0)}(y, t)$ in terms of $v(x, t)$, its derivatives, and $x^{(0)}(t)$ will be given below.

Next we (numerically) construct a linear, time-dependent transformation:

$$w = \mathbf{T}^{(0)}(t) y, \tag{4}$$

which transforms the linear part of Eq. (3) into a constant coefficient equation:

$$w = \mathbf{D}^{(0)} w + h^{(0)}(w, t), \tag{5}$$

where the form of $h^{(0)}(w, t)$ in terms of $\mathbf{T}^{(0)}(t)$ and $f^{(0)}(y, t)$ will be given below. It is the constant matrix $\mathbf{D}^{(0)}$ here that is crucial. $\mathbf{D}^{(0)}$ is a diagonal matrix with one negative eigenvalue and one positive eigenvalue, i.e. it is hyperbolic. We will show below how this hyperbolicity is inherited from the linear equation $y = \mathbf{A}^{(0)}(t) y$, which, in turn, inherits its hyperbolicity from our first guess, $x^{(0)}(t)$.

Next we consider the following integral equation associated with Eq. (5):

$$w_1(t) = \int_{t_0}^{t} e^{d_1^{(0)}(t-s)} h_1^{(0)}(w_1(s), w_2(s), s) ds,$$

$$w_2(t) = -\int_{t}^{t_L} e^{d_2^{(0)}(t-s)} h_2^{(0)}(w_1(s), w_2(s), s) ds, \tag{6}$$

where $d_1^{(0)} < 0$ is the negative eigenvalue of $\mathbf{D}^{(0)}$ and $d_2^{(0)} > 0$ is the positive eigenvalue of $\mathbf{D}^{(0)}$. It can verified by differentiation with respect to time that a solution of Eq. (6) is a solution of Eq. (5).

The transformation Eq. (4) offers several advantages.

– It facilitates the solution of the integral Eqs. (6).

– It is directly related to the definition of hyperbolicity that we use, as discussed in Appendix A.

– It provides the "first guess" for the algorithm for computing the stable and unstable manifolds of the trajectory.

The integral equation also offers several advantages. One is that it yields a solution that is defined over the entire temporal interval of interest. In this way we do not "lose data". The other is that it is ideal for numerical iteration.

It is shown in Ju et al. (2003) that if $\frac{\partial h^{(0)}}{\partial w}$ is "small", then Eq. (6) has a unique solution whose stability properties are inherited from $\mathbf{D}^{(0)}$, i.e. it is hyperbolic. One can then work backwards through the transformations and express the hyperbolic trajectory in the original x coordinates.

There are two issues here. One is that, in general, we cannot expect $\frac{\partial h^{(0)}}{\partial w}$ to be "small". The other is that we have described the procedure to compute the hyperbolic trajectory as an iterative procedure, yet we have not described how we iterate. These two issues are related in that we will show below that the iterative procedure involves refining the "first guess" $x^{(0)}(t)$ iteratively and, when the iteration is successful in finding a hyperbolic trajectory, this will force $\frac{\partial h^{(0)}}{\partial w}$ to become "small".

Now we turn to describing the procedure in detail.

3.1.2 The iterative procedure

After $n-1$ steps we obtain an approximation to a hyperbolic trajectory, denoted $x^{(n)}(t)$. The procedure is then repeated in order to converge to a hyperbolic trajectory.

Step 0: An Initial Guess

We take as our initial guess $x^{(0)}(t)$ for the iterative algorithm a path (which in general will not be a trajectory) constructed from instantaneous stagnation points (ISPs) of the
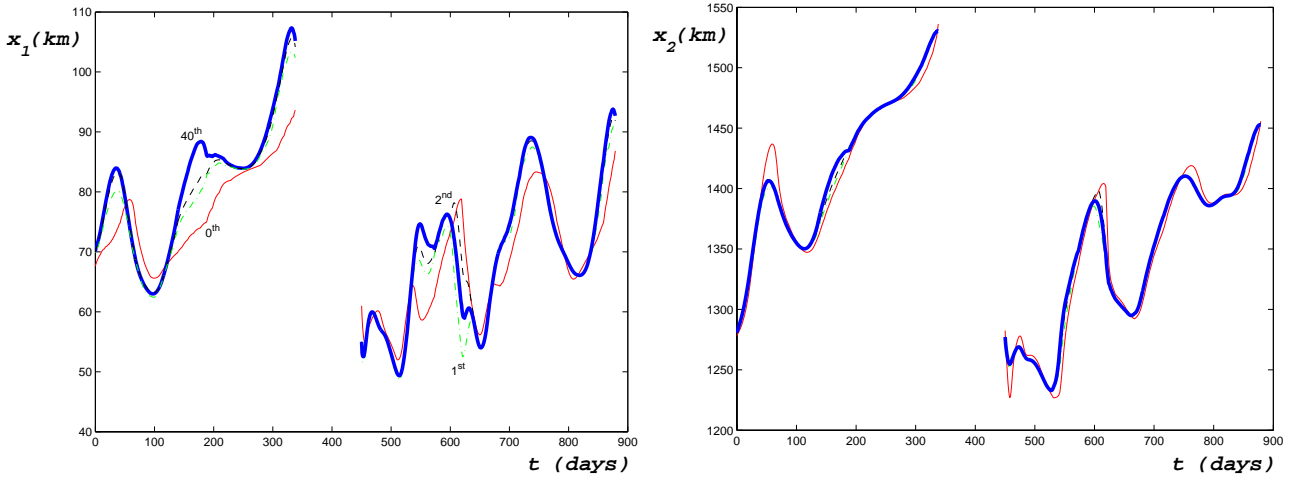
Fig. 3. Convergence of the iterative procedure to interior hyperbolic trajectories showing the initial path for the iterative procedure based on two disjoint segments of the path of stagnation points (red), the 1st iteration (green dotted-dashed line), the 2nd iteration (black line) and the converged trajectory after 40 iterations (thick blue line). Note that since the gap is not patched in the initial path, all iterations are disjoint segments (in time).
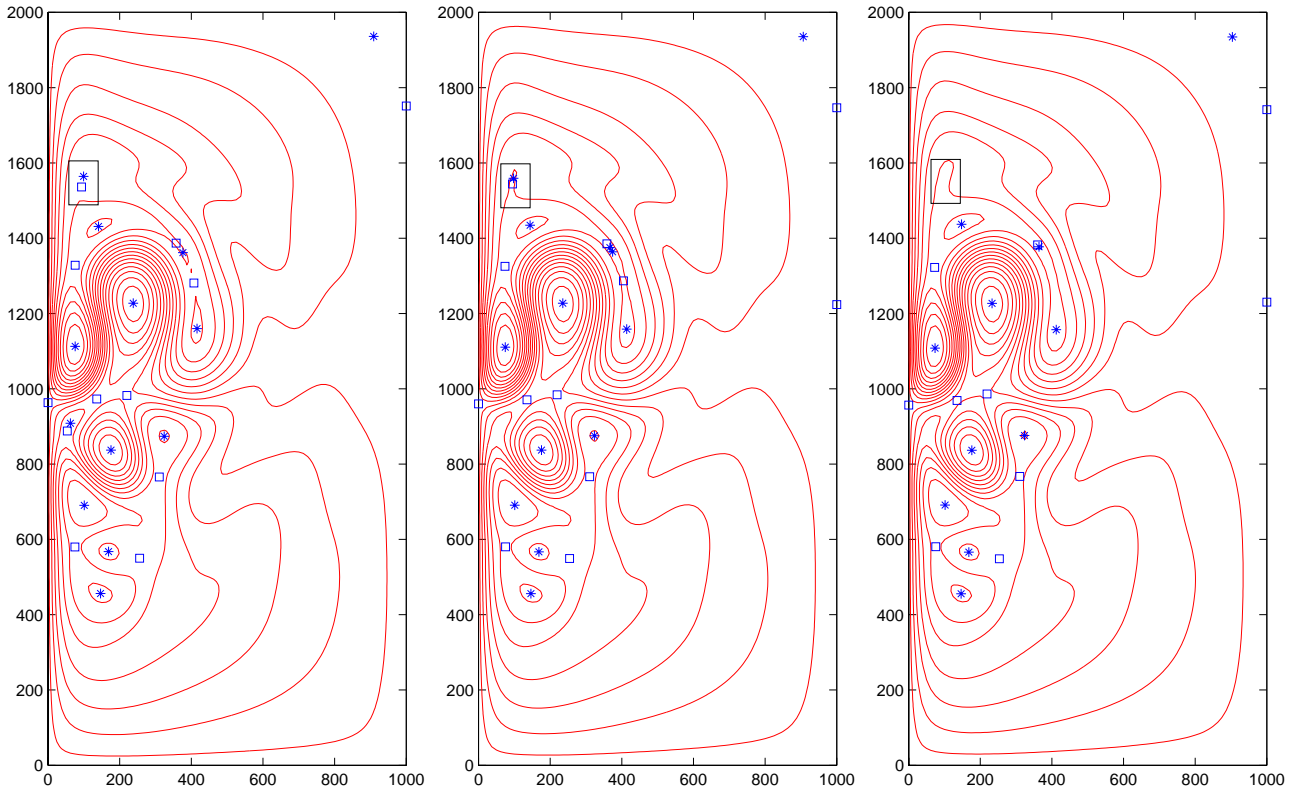

Fig. 4. The instantaneous streamlines for days 338–340 (left-to-right panels). Three successive time-slices of the data, where the boxed region contains the interior hyperbolic stagnation point shown in Fig. 3 (square) approaching an elliptic stagnation point (asterisk) and then disappearing in a saddle-node bifurcation.

Eulerian velocity field. This gives rise to two possible problems: firstly, with a finite collection of time-slices it is not always possible to be sure that two ISPs at successive times form part of the same path in the continuous system underlying our data. At present, we intervene manually in cases

of doubt, but there is scope for the development of heuristics. A second difficulty is that ISPs of the continuous problem may bifurcate, and this gives rise to gaps (in time). In some simple problems joining two different hyperbolic ISP paths by linear interpolation across the temporal gaps may
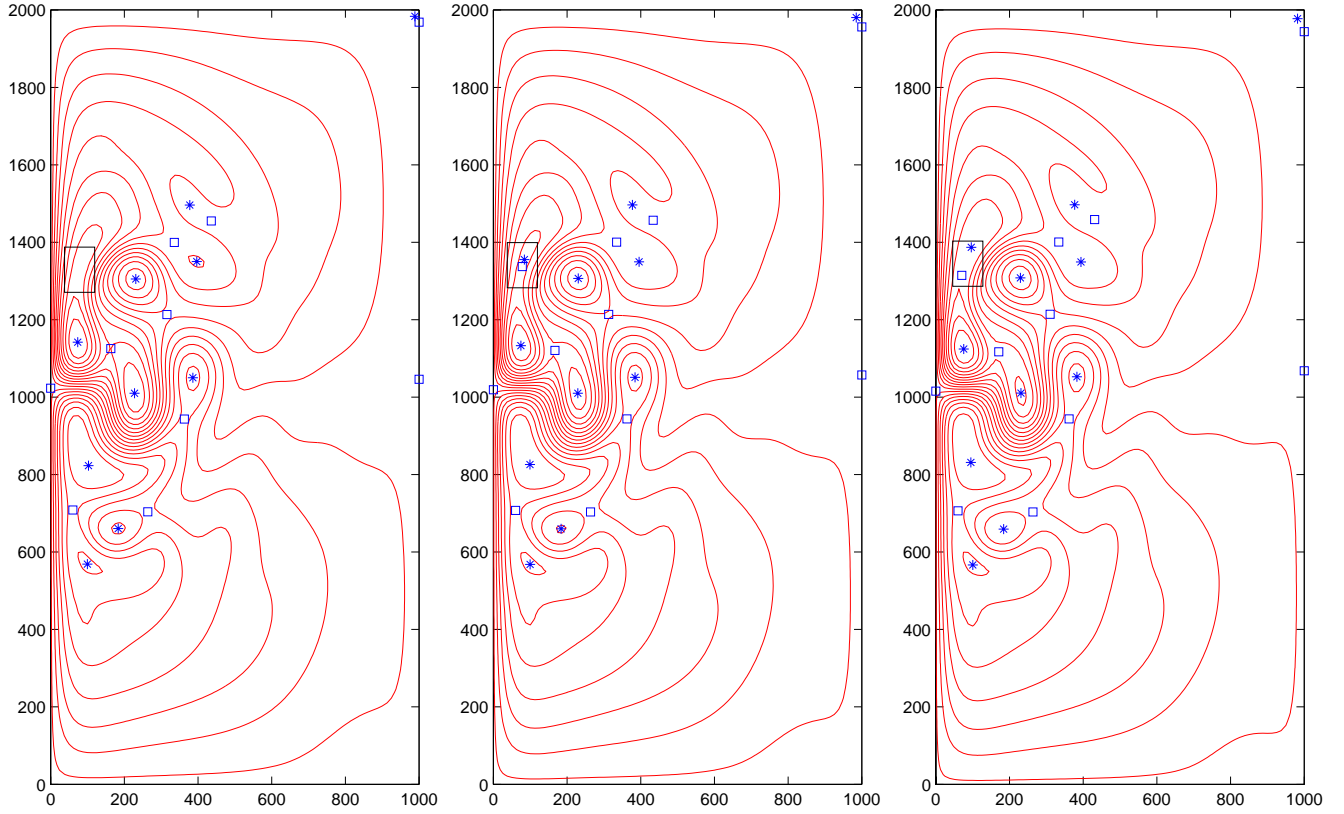
Fig. 5. The instantaneous streamlines for days 445–447 (left-to-right panels). The time-reversal of the process of Fig. 4 for the hyperbolic trajectories shown in Fig. 3. Within the boxed region hyperbolic (square) and elliptic (asterisk) stagnation points appear in a saddle-node bifurcation and then move apart.

provide a convergent initial guess, but Fig. 3 shows a gap we have not yet succeeded in patching in this manner, and Figs. 4 and 5 each show three consecutive frames of the flow field with hyperbolic and elliptic points approaching each other and vanishing (in Fig. 3 we show the bifurcating stagnation points on both sides of the bifurcation point), and the reverse time situation (in Fig. 5) on the other side of the gap. This will be explained in more detail in Sect. 4.

Note that the only relevant, and ultimate, justification for an initial guess is that the algorithm successfully converges to a hyperbolic trajectory – it is not required to have any physical interpretation in its own right.

Step 1: Localization about the Approximation to the Hyperbolic Trajectory.

Let

$$y = x - x^{(n)}(t), \tag{7}$$

Then in the y-coordinates the velocity field takes the form:

$$\dot{y} = A^{(n)}(t)y + f^{(n)}(y, t), \tag{8}$$

where

$$A^{(n)}(t) = \frac{\partial v}{\partial x}(x^{(n)}(t), t),$$

$$f^{(n)}(y, t) = v(y + x^{(n)}(t), t)y$$

$$-\frac{\partial v}{\partial x}(x^{(n)}(t), t)y - \dot{x}^{(n)}(t). \tag{9}$$

The time-dependent matrix $A^{(n)}(t)$ is required to reflect the hyperbolic nature of the path $x^{(n)}(t)$ (note that $A^{(n)}(t)$ is the Jacobian of the instantaneous velocity field evaluated on the path). This is checked in the next step.

When working on a finite time-grid, $x^{(n)}(t)$ is calculated using second-order central differences in the time direction on the interior time slices, and one-sided second-order schemes at both ends. The Jacobian matrix $A^{(n)}(t)$ is calculated using bicubic spatial interpolation of the velocity field, which typically also provides the derivatives of the velocities (see Press et al., 1992). The velocity derivatives required at the grid points are calculated using second-order central differences in space.

Step 2: Transform the Linear Part of Eq. (8) to constant coefficient and check hyperbolicity.

Solve the matrix equation

$$\dot{Y}^{(n)} = A^{(n)}(t)Y^{(n)}, \quad Y^{(n)}(t_0, t_0) = \mathbf{id}, \tag{10}$$

for the 2×2 fundamental solution matrix $Y^{(n)}(t, t_0)$, $t \in [t_0, t_L]$, with $Y^{(n)}(t_0, t_0) = \mathbf{id}$, where $\mathbf{id}$ denotes the 2×2 identity matrix.

At each instant of time compute the singular value decomposition (SVD) of $\mathbf{Y}^{(n)}(t, t_0)$:

$$\mathbf{Y}^{(n)}(t, t_0) = \mathbf{B}^{(n)}(t, t_0)e^{\Sigma^{(n)}(t,t_0)}\left(\mathbf{R}^{(n)}\right)^T(t, t_0), \qquad (11)$$

where $\mathbf{B}^{(n)}(t, t_0)$ and $\mathbf{R}^{(n)}(t, t_0)$ are orthogonal matrices, and $\Sigma^{(n)}(t, t_0)$ is a diagonal matrix. If the eigenvalues of $\frac{1}{t_L-t_0}\Sigma^{(n)}(t_L, t_0)$ are $d_1^{(n)}<0$ and $d_2^{(n)}>0$, then $x^{(n)}(t)$ is hyperbolic, and the procedure continues.

We define the constant matrix

$$\mathbf{D}^{(n)} = \frac{1}{t_L - t_0}\Sigma^{(n)}(t_L, t_0), \qquad (12)$$

and the time dependent matrix

$$\mathbf{T}^{(n)}(t) = e^{(t-t_0)\mathbf{D}^{(n)}}\mathbf{T}^{(n)}(t_0)\left(\mathbf{Y}^{(n)}\right)^{-1}(t, t_0). \qquad (13)$$

A full derivation of these two terms is given in an Appendix, together with details of the numerical procedure used to calculate them.

Then (see Ide et al., 2002, and Appendix A),

$$\boldsymbol{w} = \mathbf{T}^{(n)}(t)\boldsymbol{y}, \qquad (14)$$

transforms Eq. (8) to the form:

$$\boldsymbol{w} = \mathbf{D}^{(n)}\boldsymbol{w} + \boldsymbol{h}^{(n)}(\boldsymbol{w}, t), \qquad (15)$$

where $\boldsymbol{h}^{(n)}(w, t)$ is given by:

$$\boldsymbol{h}^{(n)}(\boldsymbol{w}, t) = \mathbf{T}^{(n)}(t)\boldsymbol{f}\left(\left(\mathbf{T}^{(n)}\right)^{-1}(t)\boldsymbol{w}, t\right). \qquad (16)$$

Step 3: Transform Eq. (15) to integral equation form and compute an approximate solution.

The improvement to the approximate hyperbolic trajectory obtained in the previous step is obtained by solving an integral equation (Eq. 6) by the iteration:

$$w_1^{(n+1)}(t) = \int_{t_0}^{t} e^{d_1^{(n)}(t-s)}\tilde{h}_1^{(n)}(w_1^{(n)}(s), w_2^{(n)}(s), s)ds,$$

$$w_2^{(n+1)}(t) = -\int_{t}^{t_L} e^{d_2^{(n)}(t-s)}\tilde{h}_2^{(n)}(w_1^{(n)}(s),$$

$$w_2^{(n)}(s), s)ds, \quad (17)$$

where $\boldsymbol{h}^{(n)}(\boldsymbol{w}^{(n)}(t), t)$ is given by

$$\boldsymbol{h}(\boldsymbol{w}^{(n)}(t), t) = -\mathbf{T}^{(n)}(t)\boldsymbol{x}^{(n)}(t), \qquad (18)$$

i.e. the nonlinear terms in $\boldsymbol{h}^{(n)}(\boldsymbol{w}, t)$ are neglected. In this case, since $\boldsymbol{x}^{(n)}(t)$ is a known function from the previous step in the iteration, $\boldsymbol{w}^{(n+1)}(t)$ is obtained by a quadrature. These integrals are calculated numerically using the trapezium rule after a change of variables $\tau = e^{d_i^{(n)}(t-s)}$ is introduced to extract the exponential part of the behaviour.

The neglect of the nonlinear terms in $\boldsymbol{h}^{(n)}(w, t)$ is an important point. A consequence is that a solution of Eq. (17) is not a trajectory of Eq. (8). In order to remedy this situation we will need to require that the nonlinear terms vanish in the

limit of an infinite number of iterations. We will discuss this shortly.

We have now gone through one step of the iterative procedure. To obtain the approximation to the hyperbolic trajectory to be used in the next step we transform back to the original $\boldsymbol{x}$-coordinates by:

$$\boldsymbol{x}^{(n+1)}(t) = \boldsymbol{x}^{(n)}(t) + \left(\mathbf{T}^{(n)}\right)^{-1}(t)\boldsymbol{w}^{(n+1)}(t), \qquad (19)$$

and repeat the procedure with $\boldsymbol{x}^{(n+1)}(t)$.

3.1.3 Convergence of the Iterative Procedure

In order for the iterative procedure to be successful in yielding a hyperbolic trajectory the iterative solution of the integral Eq. (17) must yield a fixed point, $w^{(\infty)}(t)$, with

$$\lim_{n\to\infty}\left\|\mathbf{T}^{(n)}(t)\left(\boldsymbol{v}\left(\left(\mathbf{T}^{(n)}\right)^{-1}(t)\boldsymbol{w}^{(n)}(t) + \boldsymbol{x}^{(n)}(t), t\right)\right.\right.$$

$$\left.\left. -\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}\left(\boldsymbol{x}^{(n)}(t), t\right)\right)\right\| = 0, \qquad (20)$$

in some convenient norm.

In order for the trajectory to be hyperbolic we must have:

$$\lim_{n\to\infty} d_1^{(n)} < 0, \qquad \lim_{n\to\infty} d_2^{(n)} > 0. \qquad (21)$$

Each of these quantities is monitored at each step until a numerically acceptable tolerance for convergence is reached (10^{-5} or better).

3.2 Computation of the stable and unstable manifolds of the hyperbolic trajectory

Once a hyperbolic trajectory has been found, we want to compute its stable and unstable manifolds. We first give an overview of the method used in Mancho et al. (2003), as well as describe the general issues involved.

3.2.1 Overview of the method and issues associated with computing the stable and unstable manifolds of a hyperbolic trajectory

We discuss the computation of the unstable manifold of the hyperbolic trajectory. The discussion of the computation of the stable manifold is completely analogous, the only exceptions being that we start the computation at the end of the data set (i.e. t_L), and evolve backwards in time to the beginning of the data set.

We consider L time increments between t_0 and t_L, i.e.

$$t_0 < t_1 < t_2 < \cdots < t_{L-1} < t_L, \quad t_i < t_j, \, i < j, \qquad (22)$$

where the increments need not be equally spaced. These are the times at which we wish to observe, or "output", the manifold.

The idea for computing the unstable manifold is simple. We begin with a small initial segment of the unstable manifold, either centered on the hyperbolic trajectory, or with one endpoint being the hyperbolic trajectory, at the beginning of the data set. This segment is then evolved in time to the end of the data set. However, since the initial segment is represented by discrete points, the process of evolving this segment to the end of the data set gives rise to numerous numerical issues, that we now describe.

– The initial segment of the unstable manifold

We shall take as the initial guess a "small" segment of the linearization of the unstable manifold of the hyperbolic trajectory at $t=t_0$. The stable and unstable manifold theorem for hyperbolic trajectories (Mancho et al., 2003) tells us that this is a good approximation for a small enough segment (which will, generally, be a problem dependent notion).

This initial segment of the linearized unstable manifold is provided to us by the algorithm for computing hyperbolic trajectories given above. Recall that the matrix $\mathbf{D}$ had a positive eigenvalue in the second column, then the associated linearized unstable direction at $t=t_0$ in the w-coordinates is simply $e_2=(0, 1)^T$, which corresponds to $\mathbf{T}^{-1}(t_0)e_2$ in the y-coordinates.

Along this direction the initial segment of the manifold is realized by taking five closely spaced points, where the middle point corresponds to the hyperbolic trajectory (if the hyperbolic trajectory is on a solid boundary, we also take five points with the end point being on the hyperbolic trajectory).

This initial segment is then evolved forward in time. As the points which make up this computational representation of the manifold evolve they may grow apart, giving rise to unacceptably large gaps between adjacent points on the manifold. In order to address this a precise formulation of the acceptability of gaps, and some means of filling them is necessary.

It is clear that any such method of "filling gaps" amounts to a claim about the behaviour of the manifold between points at a given time. Addressing this issue requires some form of interpolation.

– Interpolation

In Mancho et al. (2003) several forms of interpolation (e.g. linear, Lagrange polynomials, cubic splines) were compared for use in the manifold computation algorithm and it was found that a method due to Dritschel (1989) developed in the course of his work on contour advection gave the best results (according to various criteria discussed in Mancho et al. (2003)).

Dritschel's method makes use of the positions of the four consecutive points x_{j-1}, x_j, x_{j+1} and x_{j+2} to represent the curve between points x_j and x_{j+1} as

$$x(p) = x_j + p\, t_j + \eta_j(p)\, n_j, \tag{23}$$

for $0 \le p \le 1$ with $x(0)=x_j$ and $x(1)=x_{j+1}$, where

$$t_j = (a_j, b_j) = x_{j+1} - x_j, \qquad t_j \in \mathbb{R}^2 \tag{24}$$

$$n_j = (-b_j, a_j), \qquad n_j \in \mathbb{R}^2 \tag{25}$$

$$\eta_j(p) = \mu_j p + \beta_j p^2 + \gamma_j p^3, \qquad \eta_j(p) \in \mathbb{R}. \tag{26}$$

The cubic interpolation coefficients μ_j, β_j and γ_j are

$$\mu_j = -\frac{1}{3} d_j \kappa_j - \frac{1}{6} d_j \kappa_{j+1}, \tag{27}$$

$$\beta_j = \frac{1}{2} d_j \kappa_j, \tag{28}$$

$$\gamma_j = \frac{1}{6} d_j(\kappa_{j+1} - \kappa_j), \tag{29}$$

where

$$d_j = |x_{j+1} - x_j|, \tag{30}$$

and

$$\kappa_j = 2 \frac{a_{j-1} b_j - b_{j-1} a_j}{|d_{j-1}^2 t_j + d_j^2 t_{j-1}|} \tag{31}$$

is the local curvature defined by a circle through the three points, x_{j-1}, x_j, and x_{j+1}. Therefore, the representation $x(p)$ for $0 \le p \le 1$ uses the local curvatures at each end of its interval to describe the smooth curve between x_j and x_{j+1}.

– A measure of the density of points along the computed manifold

Following Dritschel (1989) and Dritschel and Ambaum (1997), for each pair of consecutive points x_j and x_{j-1} in manifold at a given time we calculate a desired density of points, ρ_j, defined by

$$\rho_j \equiv \frac{(\bar{\kappa}_j L)^{\frac{1}{2}}}{\mu L} + \bar{\kappa}_j, \tag{32}$$

or $2/\delta$, whichever is smaller (so that nodes cannot get closer than $\delta/2$, where δ is a problem dependent distance which serves as a small-scale cut-off distance for resolving manifold details). The parameter μ in this equation ultimately controls the overall point density along the manifold. Small values of μ correspond to a high point density, but the parameter needs tuning for individual problems.

The quantity $\bar{\kappa}_j$ in Eq. (32) is defined in terms of $\check{\kappa}$,

$$\bar{\kappa}_j \equiv (\check{\kappa}_j + \check{\kappa}_{j+1})/2, \tag{33}$$

which in turn is defined by

$$\check{\kappa}_j = \frac{w_{j-1} \tilde{\kappa}_{j-1} + w_j \tilde{\kappa}_j}{w_{j-1} + w_j}, \tag{34}$$

which uses the weighting $w_j = d_j/(d_j^2 + 4\delta^2)$ and the further curvature $\tilde{\kappa}_j$, which itself is defined by

$$\tilde{\kappa}_j = \sqrt{\kappa_j^2 + 1/L^2}, \tag{35}$$

where κ_j, finally, is the local curvature defined in Eq. (31), and L is defined in Dritschel and Ambaum (1997) to be a

length typical of the large-scale vorticity distribution; in the absence of such a scale we substitute a typical length of the domain for the calculation.

– A criterion for unacceptably large gaps

Now we can define what we mean by an acceptable gap between adjacent points on the manifold. The distance d_j between x_j and x_{j+1} is unacceptable if $\sigma_j > 1$, where σ_j is defined by $\sigma_j = \rho_j d_j$.

3.2.2 The algorithm

The computation of the unstable manifold proceeds as follows.

Step1: Evolve the approximate initial segment of the unstable manifold forward in time, with for instance a 5th order Runge-Kutta method, starting from $t = t_0$.

Step 2: At each observation time, t_k, check the size of gaps between adjacent points on the manifold.

At each time t_k on the time grid of the computation we check each pair of adjacent points with the criterion described above.

Step 3: Apply a point insertion and interpolation scheme at time t_{k-1}.

If the gap is unacceptably large we insert a new point at the previous observation time t_{k-1} between the positions of the points at time t_{k-1} that evolve along trajectories to the points x_j and x_{j+1} at time t_k using Eq. (23) to locally represent the manifold at t_{k-1}. We insert points at a time before a gap becomes unacceptably large so that the curve used for interpolation is always an acceptable representation of the manifold, by our definition, and we evolve those points with a 5th order Runge-Kutta method to time t_k.

This procedure is carried out for each pair of adjacent points making up the manifold at t_k. If new points are added, the entire procedure is repeated again.

The procedure is iterated until there are no gaps exceeding the tolerance at time t_k.

Step 4: Apply point redistribution at t_k.

Once the gap size acceptability criterion is satisfied at a time t_k we use the point redistribution algorithm described in Dritschel (1989) in an attempt to ensure that points are removed from less demanding parts of the manifold and thereby reduce the total number of points included in the manifold.

During redistribution the end points of the manifold are held fixed. Let n be the number of nodes at t_k. Compute

$$q = \sum_{j=1}^{n} \sigma_j \qquad (36)$$

and define $\tilde{n} = [q] + 2$ (i.e. two more than the nearest integer to q). The $n-2$ "old" nodes between the end points will be replaced by $\tilde{n} - 1$ entirely new nodes in such a way that the spacing of new nodes is approximately consistent with the desired average density, controlled by the parameter μ.

Let $\sigma'_j = \sigma_j \tilde{n}/q$ so that $\sum_{j=1}^{n} \sigma'_j = \tilde{n}$. Then, the positions of the new nodes $i = 2, ..., \tilde{n}$ are found successively by seeking j and p such that,

$$\sum_{l=1}^{j-1} \sigma'_l + \sigma'_j p = i - 1, \qquad (37)$$

and placing each new node i between the old nodes j and $j+1$ at the position $x(p)$ given in Eq. (23).

Iterative procedure: evolve in time, applying this procedure at each observation time until the end of the data set is reached, $t = t_L$.

Once the point insertion and interpolation scheme, followed by point redistribution, has been completed for the manifold at t_k, the entire manifold is evolved to the next observation time, t_{k+1}. Then the point insertion and interpolation scheme and point redistribution is applied at this time. The procedure is then repeated at each observation time until the end of the time interval of interest is reached.

4 Application of the procedure to a data set obtained from a quasigeostrophical model.

We now apply our procedure to the data from the quasigeostrophic model described in Sect. 2

To begin, we compute the hyperbolic trajectories on the boundaries. As explained above these computations consist of an iterative procedure which starts from an initial guess based on the path of instantaneous stagnation points. Figures 1 and 2 show the ISPs for the western and eastern boundaries, respectively. In order to show the bifurcations of the stagnation points circles are used for the stable points ("stable" in the instantaneous velocity field, that is) and crosses are for the unstable points.

Bifurcations of ISPs complicate the choice of the initial guess: on the western boundary we have two coexisting ISPs for a time, while on the eastern boundary there is a time interval for which the flow has no ISPs. To deal with the bifurcation on the western boundary an initial guess (green path) was used, with some points of the ISP path replaced by points calculated from a cubic polynomial that smoothly matches the two branches of the ISPs between days 396 to 406 exclusive (see Fig. 1). On the eastern boundary the small gap between the two saddle-node bifurcations has been patched with a constant value (see the green line in Fig. 2). The hyperbolic trajectories obtained by successive iterations of the algorithm are displayed in red and black lines in those figures.

Convergence of the algorithm is reached after a different number of iterations for each side (i.e. for the western and eastern boundaries), however the accuracy of the hyperbolic trajectory is reduced in the time intervals where the ISPs bifurcate. In Fig. 1 from day 400 to day 430 the converged solution (black line) obtained after 50 iterations does not coincide with the trajectory that would be computed by an integration starting from any point of the trajectory at a previous
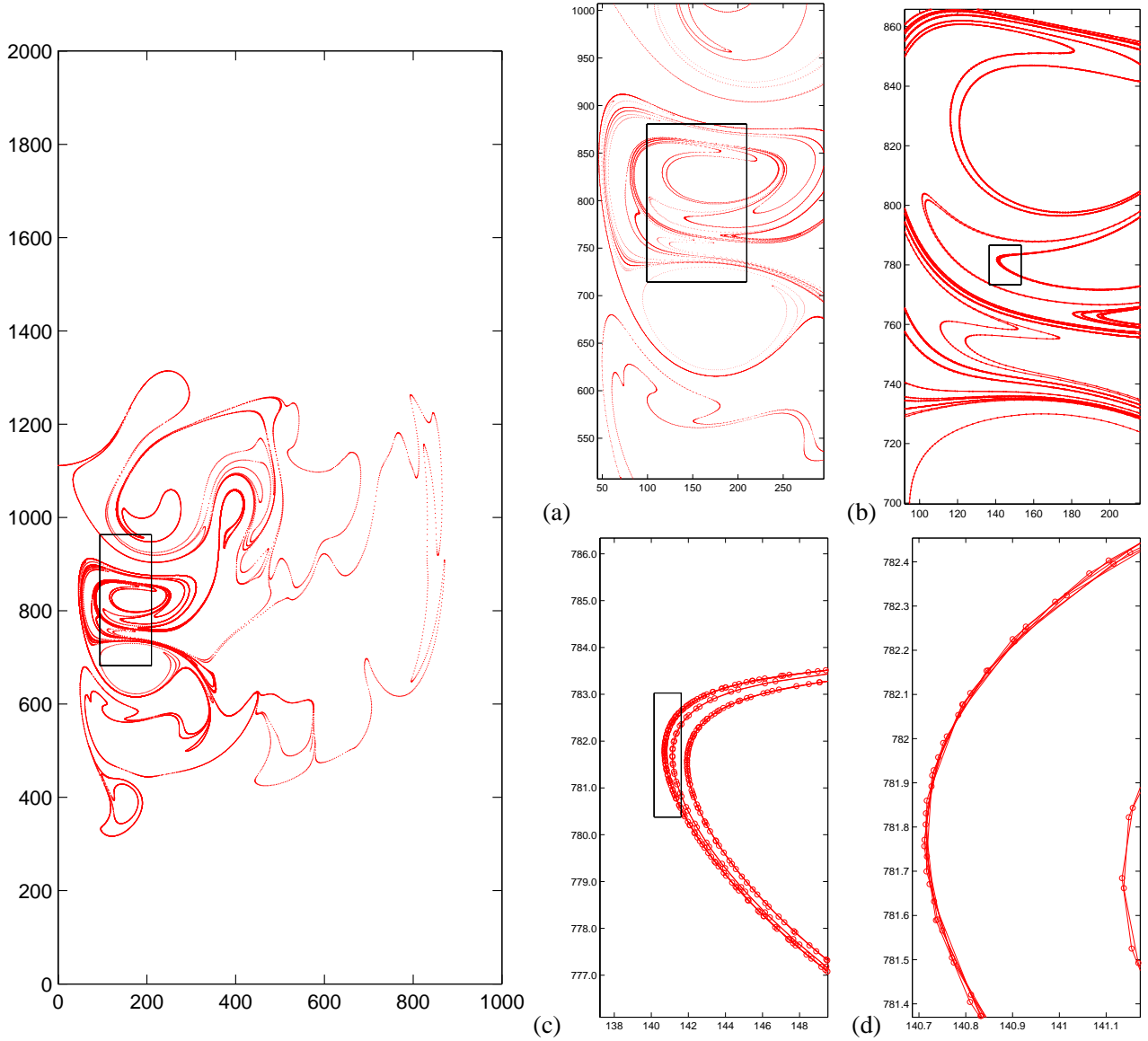
Fig. 6. The unstable manifold (at day 430) grown from the western boundary starting at day 50. Smaller figures show successive magnifications of the manifold. The length of the manifold shown is composed of 51 611 points.

time, and in Fig. 2 the black line displays a slight spike just where the gap has been filled. At the places where the bifurcations occur there is a sudden jump in the path (smoothed slightly by fitting a cubic profile) which induces a corresponding jump in the coordinate system used to localise the system, and thus a spike in the $x^{(n)}$ term in Eq. (9). The system remains sensitive to these spikes and the final "trajectory" for such cases still shows traces of them. This does not happen with smooth initial paths (e.g. those from non-bifurcating ISPs).

Figure 3 shows the x_1 and x_2 coordinates of two interior ISPs which bifurcate at days 339 and 446 (see Figs. 4 and 5). As discussed in Sect. 3.1.2 those gaps are not filled, since in that case we have not found a patching procedure for which the iterative procedure converges, and therefore we compute

the interior hyperbolic trajectories only for the time intervals displayed in Fig. 3, which again shows a sequence of iterations.

Once we have the hyperbolic trajectories, we proceed with the computation of their unstable and stable manifolds. To begin with we need a small initial segment of the unstable or stable manifolds. As discussed in Sect. 3.2.1 the algorithm to compute the hyperbolic trajectories also provides a subspace tangent to the stable and unstable manifolds. For the case of hyperbolic trajectories on the western and eastern boundaries this is not necessary as it can easily be shown that the required linear segments of the unstable and stable manifolds are perpendicular to the boundaries. Equation (1) on the western boundary can be linearized around the hyperbolic
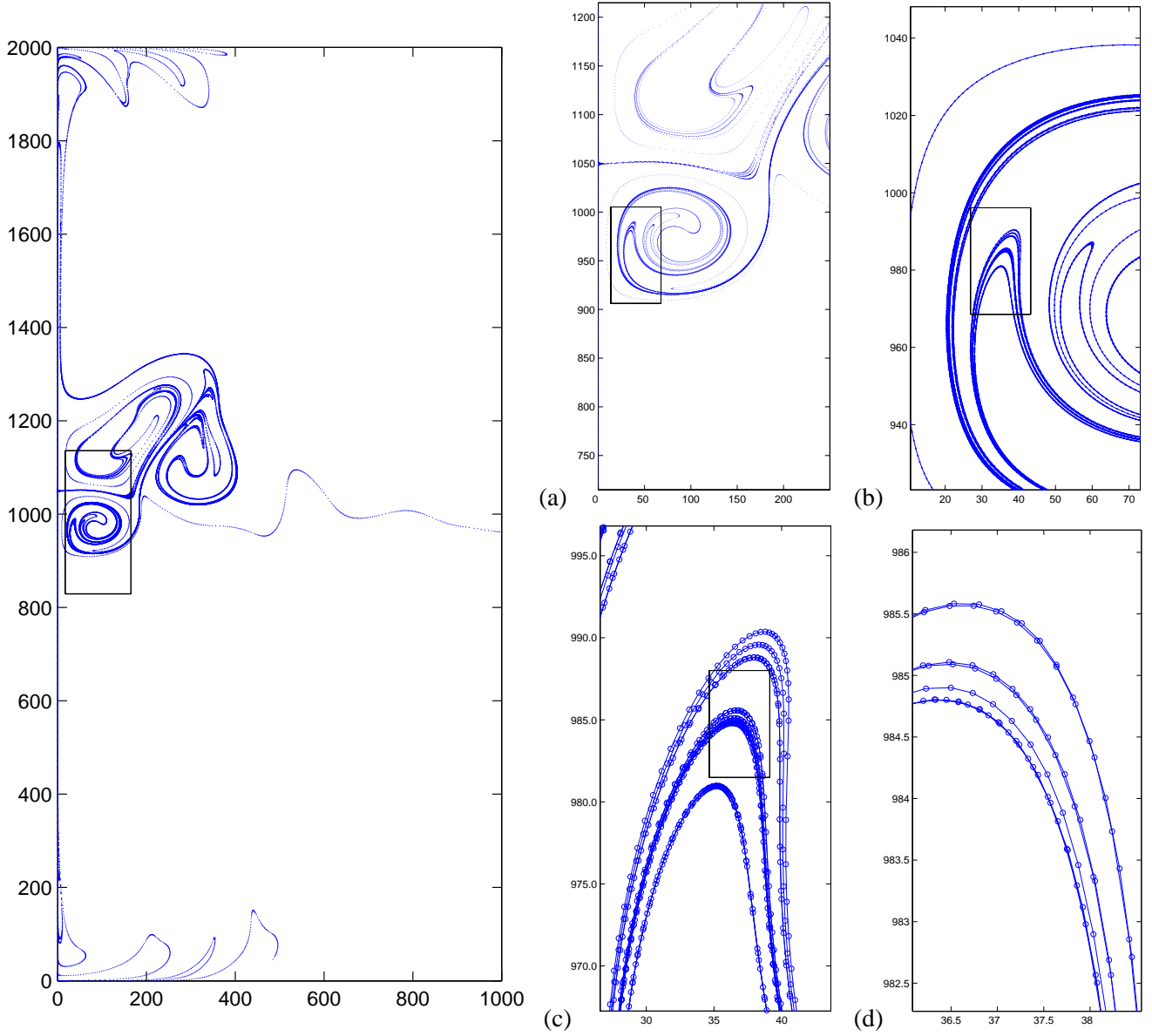
Fig. 7. The stable manifold (at day 380) grown from the eastern boundary starting at day 1495, with successive magnifications. The length of the manifold shown is composed of 33 413 points.

trajectory, $x = x_{\text{hyp}}(t) + x'$,

$$\frac{d}{dt} \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \begin{pmatrix} \frac{\partial v_1}{\partial x_1} & \frac{\partial v_1}{\partial x_2} \\ \frac{\partial v_2}{\partial x_1} & \frac{\partial v_2}{\partial x_2} \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}. \tag{38}$$

Now $\frac{\partial v_1}{\partial x_2} = 0$, since v_1 is identically zero on the western boundary and $\frac{\partial v_2}{\partial x_1} = 0$, since we impose a no-stress condition on the boundary in the quasigeostrophic model. On the other hand incorporating the divergence-free condition $\frac{\partial v_1}{\partial x_1} = -\frac{\partial v_2}{\partial x_2}$ into Eq. (38) gives

$$\frac{d}{dt} \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix} = \frac{\partial x_2}{\partial v_2} \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x'_1 \\ x'_2 \end{pmatrix}. \tag{39}$$

Since $\frac{\partial x_2}{\partial v_2}(x_{\text{hyp}}(t)) < 0$, i.e. v_2 is a decreasing function on the western boundary close to the hyperbolic trajectory, it is

clearly seen that the unstable subspace is perpendicular to the boundary which contains the stable manifold. An analogous argument shows that the stable subspace on the eastern boundary is perpendicular to the boundary. In order to preserve this perpendicularity in the numerical model it is important that the spatial interpolation maintains the no-stress condition for the velocity field at the boundaries.

Once we have the initial segments we compute the manifolds to the boundary and interior hyperbolic trajectories using the methods described in Sect. 3.2.1. To perform the integration we have rescaled the domain $[0, 1000] \times [0, 2000]$ to $[0, 1] \times [0, 2]$ to get a domain similar to that used in Mancho et al. (2003) and we have accordingly used the same
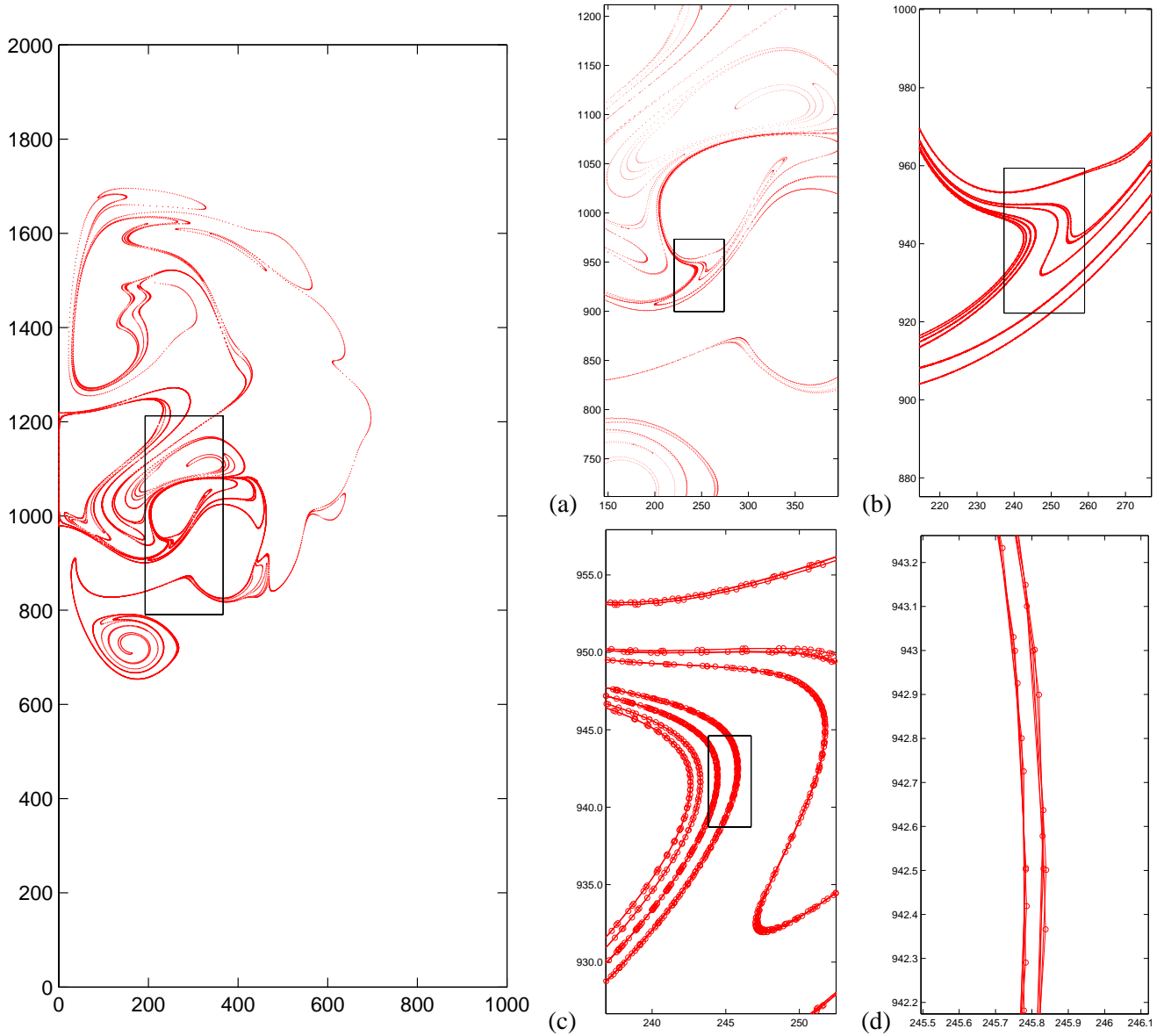
Fig. 8. The unstable manifold (at day 400) grown from the interior hyperbolic trajectory starting at day 120 shown in Fig. 3 with successive magnifications. The length of the manifold shown is composed of 34 220 points.

parameter values:

$$L = 3, \ \delta = 10^{-6}, \ \mu = 0.005.$$

Figure 6 shows a long unstable manifold associated with a hyperbolic trajectory on the western boundary and successive magnifications showing how the density of points changes to resolve high curvatures in the manifold, of which we compute spatial features in the range of hundreds of meters. The same is shown in Fig. 7 for the stable manifold of the hyperbolic trajectory on the eastern boundary.

In Fig. 8 we show the result of the unstable manifold grown for an interior hyperbolic trajectory. The hyperbolic trajectory chosen is the one computed over the time interval from day 0 to day 339. The unstable manifold is shown in the figure at day 400, and the computation of the unstable

manifold is begun at day 120. Note that the unstable manifold is evolved outside the time interval for which the hyperbolic trajectory is originally computed. In Fig. 9 we show the stable manifold grown from an interior hyperbolic trajectory. The hyperbolic trajectory chosen is the one computed over the time interval from day 446 to day 880. The stable manifold is shown in the figure at day 380, and the computation is begun at day 595. Note also that the stable manifold is evolved outside the time interval for which the hyperbolic trajectory is originally computed.

An obvious question is whether or not the hyperbolic trajectory computed on the time interval from day 0 to day 339 is the same as the hyperbolic trajectory computed on the time interval from day 446 to 880 in the sense that if the trajectory were to evolve in time beyond day 339 it would smoothly
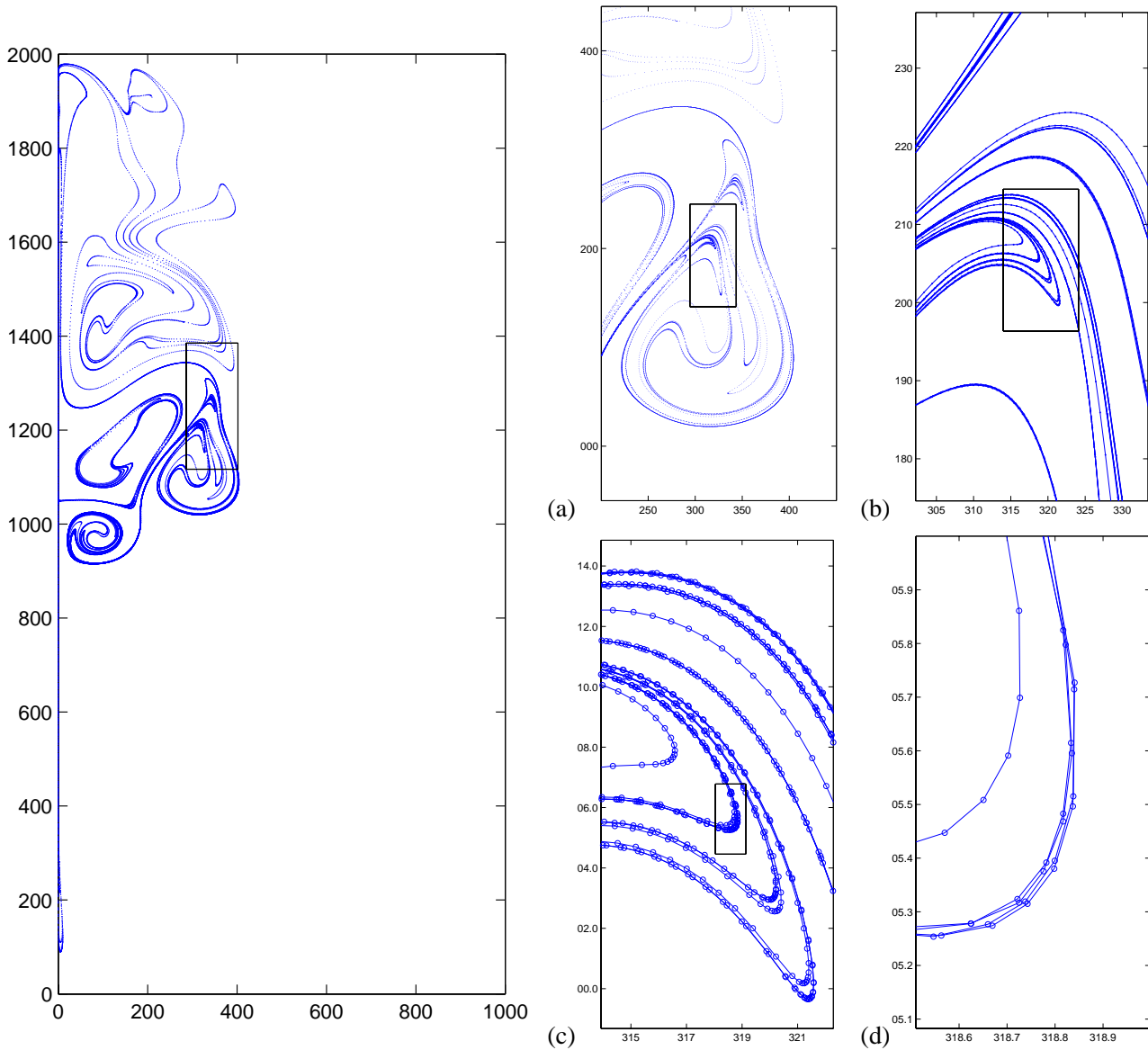
Fig. 9. The stable manifold (at day 380) grown from the interior hyperbolic trajectory starting at day 595 shown in Fig. 3. Smaller figures show successive magnifications of the manifold. The length of the manifold shown is composed of 45 509 points.

join up with the other trajectory at day 446. Answering this question is beyond the scope of this paper, but significant additional numerical work could provide an answer, as well as yield an improvement to the iterative method for computing hyperbolic trajectories as well as more insight into the consequences for hyperbolic trajectories when ISPs bifurcate.

As we discussed in the introduction, there is no problem evolving the stable or unstable manifold outside the time interval where the hyperbolic trajectory, whose stable and unstable manifold we are computing, was found with the iterative method. We are simply evolving a material curve, and the original hyperbolic trajectory must remain somewhere in this curve. In our situation, if the two hyperbolic trajectories defined on non-overlapping time intervals actually did lie on the same trajectory then we expect the trajectory to be in the

intersection of the stable manifold computed for the hyperbolic trajectory in the time interval from day 446 to day 880 with the unstable manifold computed for the hyperbolic trajectory in the time interval from day 0 to day 339 on any day between 339 and 446. Numerically, it is difficult to establish that these intersection points lie on these trajectories because of the saddle point nature of the trajectories, i.e. errors in conditions grow exponentially. A way to proceed might be to construct a new initial guess for the iterative procedure for finding hyperbolic trajectories by spanning the gap between day 339 and day 446 with a curve of initial conditions given by the appropriate intersections of the stable manifold of the hyperbolic trajectory originally computed for days 446 to 880, but evolved backwards in time across this gap, with the unstable manifold of the hyperbolic trajectory originally
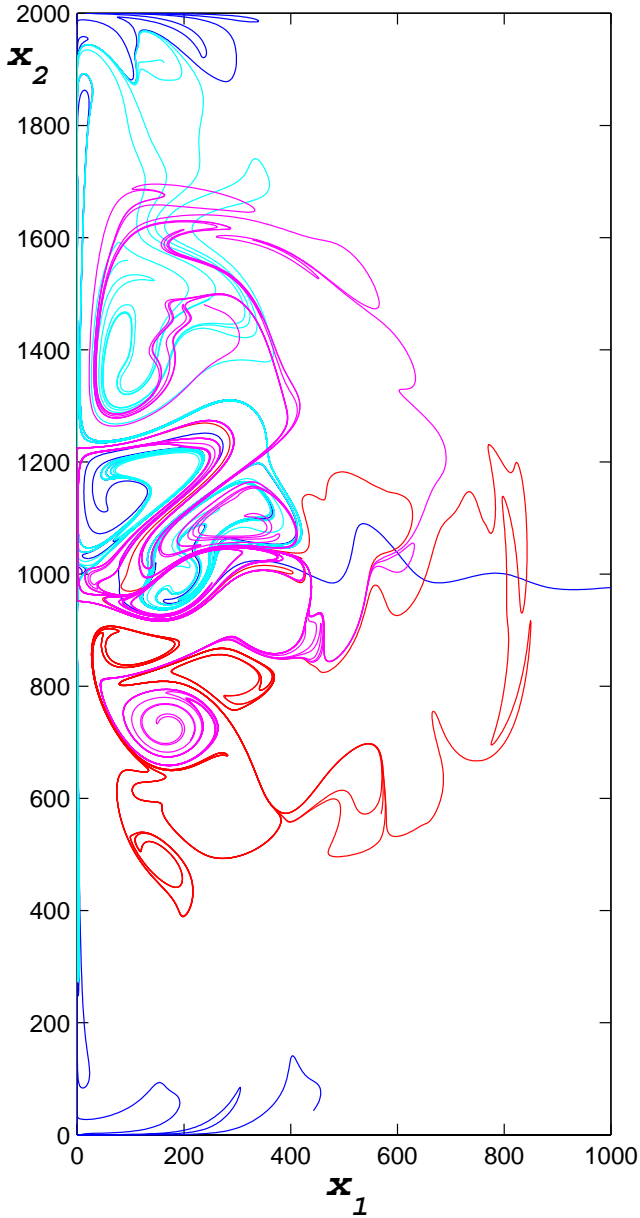
Fig. 10. Coexisting manifolds at day 395: The unstable manifold from the western boundary (red, 17 098 points), the unstable manifold of the interior hyperbolic trajectory (magenta, 27 133 points), the stable manifold of the interior hyperbolic trajectory (cyan, 21 521 points) and eastern boundary stable manifold (blue, 16 078 points).

computed for days 0 to 339, but evolved forwards in time across this gap. This is a numerically intensive project which will be left for future work.

Figure 10 shows all the manifolds that we computed on day 395.

The redistribution procedure is very useful for this computation. In a previous study (Mancho et al., 2003) of the Duffing equation the performance was similar to other manifold computation techniques, but in this data set, where the time-

dependence of the vector field varies greatly between different spatial regions, trajectories on the manifold can not only move apart to leave gaps, but also (as Fig. 11d shows) accumulate in regions where they are not needed for an accurate representation of the manifold. Figure 11c shows how the redistribution algorithm reduces the point density in the less curved parts of the manifold. Figure 11f shows how over-resolution can generate jaggedness in the folds of the manifold. This is also prevented by redistribution (see Fig. 11e).

Appendix A

Finite time hyperbolicity and the construction of the linear, time-dependent coordinate transformation to constant coefficients

Hyperbolicity of the path $x^{(n)}(t)$ (which may, or may not, be a trajectory) follows from the properties of the linearization of the velocity field about $x^{(n)}(t)$, $y=A^{(n)}(t)y$ (see Eqs. 7, 8 and 9). For notational simplicity, in the following we will leave the superscript (n) off the various quantities.

Let $Y(t, t_0)$ denote the fundamental solution matrix of the linear equation $y=A(t)y$, i.e. it is a 2×2 matrix whose columns are linearly independent solutions of $y=A(t)y$ and it satisfies $Y(t_0, t_0)=\text{id}$.

The finite time Lyapunov exponents are the logarithms of the eigenvalues of the symmetric matrix:

$$\left(Y^T(t_L, t_0)Y(t_L, t_0)\right)^{\frac{1}{2(t_L-t_0)}}, \tag{A1}$$

and we say that a path is hyperbolic on the finite time interval $t_0 \leq t \leq t_L$ if the finite time Lyapunov exponents on this time interval are nonzero. The path is saddle-like in stability (the case of interest to us), if one finite time Lyapunov exponent is positive and the other negative (this generalizes naturally to higher dimensions, but for our purposes we restrict ourselves to two dimensions). This definition of finite time Lyapunov exponent has been in use in the predictability community for some time, see Lapeyre (2002), Legras and Vautard (1996), and Farrell and Ioannou (1996). In the ordinary differential equations community characterization of hyperbolicity is more often given in terms of exponential dichotomies. In our context the approaches are equivalent, as is shown in Ide et al. (2002), see also Dieci et al. (1997).

The finite time Lyapunov exponents can also be characterized in terms of the singular value decomposition (SVD) of $Y(t, t_0)$ (Greene and Kim, 1987; Geist et al., 1990; Farrell and Ioannou, 1996; Legras and Vautard, 1996; Dieci et al., 1997; Lapeyre, 2002). This will prove useful to us when we derive the transformation that makes the coefficients of the linear part of the velocity field constant.

In terms of its SVD, we can write $Y(t, t_0)$ as:

$$Y(t, t_0) = B(t, t_0)e^{\Sigma(t, t_0)}R^T(t, t_0), \tag{A2}$$

where $B(t, t_0)$ and $R(t, t_0)$ are time-dependent orthogonal matrices, i.e. $B^T(t, t_0)B(t, t_0)=B(t, t_0)B^T(t, t_0)=\text{id}$,
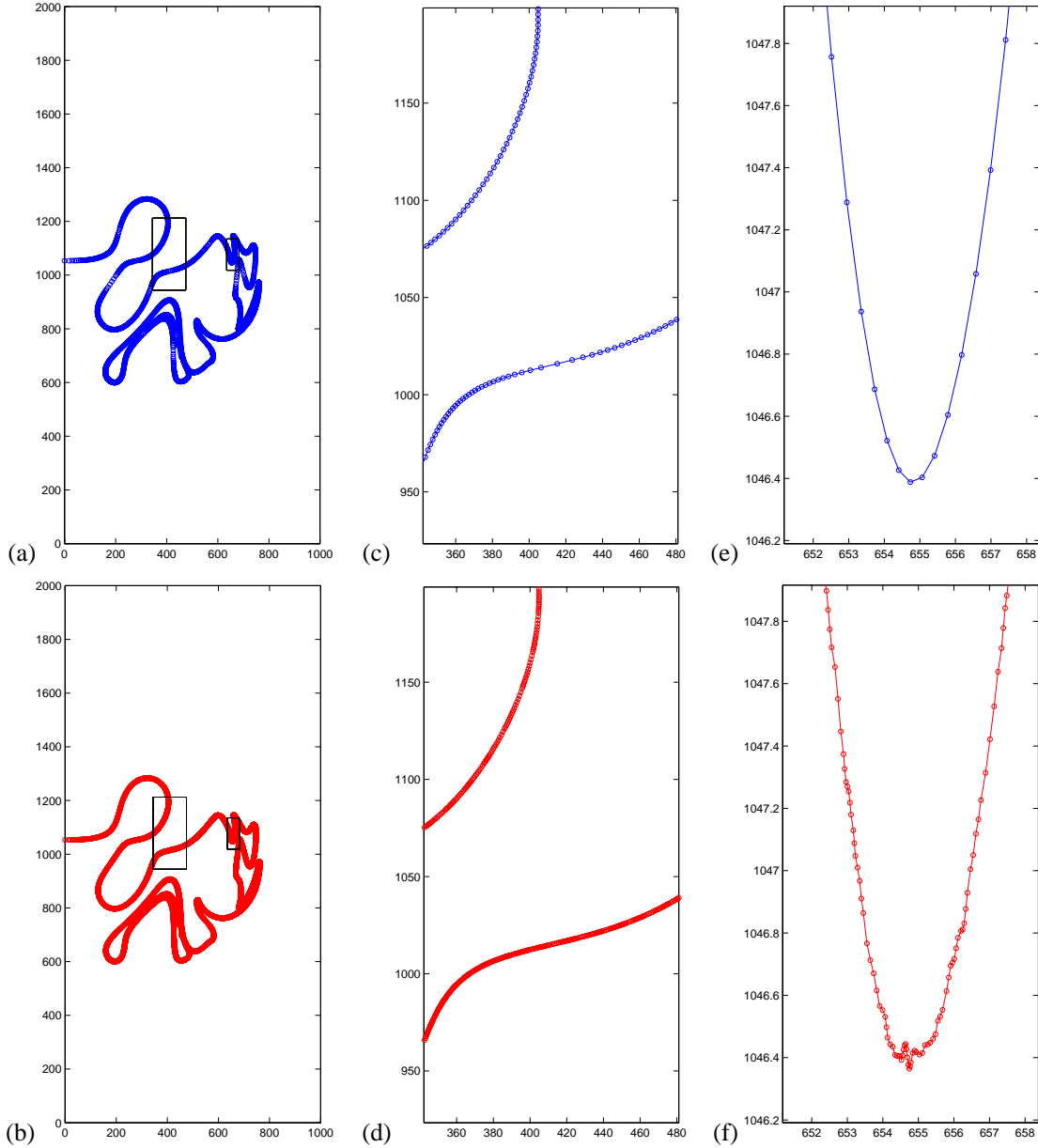
Fig. 11. The unstable manifold (at day 290) grown from the western boundary hyperbolic trajectory since day 50. **(a)** The global appearance of the manifold obtained with the algorithm using redistribution. The total number of points on it at this stage is 2309. **(b)** The global appearance of the manifold obtained with the algorithm without using redistribution. The total number of points on it at this stage is 7660. **(c)** A magnification of the zone marked with a big rectangle, for the manifold computed with redistribution; **(d)** the same as in (c) without redistribution. **(e)** A magnification of the zone marked with a small rectangle for the manifold computed with redistribution; **(f)** the same as in (e) without redistribution.

$\mathbf{R}^T(t, t_0)\mathbf{R}(t, t_0) = \mathbf{R}(t, t_0)\mathbf{R}^T(t, t_0) = \mathbf{id}$ and $\Sigma(t, t_0)$ is a diagonal matrix. It then follows that the diagonal elements of $\frac{1}{t_L - t_0} \Sigma(t_L, t_0)$ are the finite time Lyapunov exponents as defined above.

Now, for completeness, we provide a brief derivation of the transformation of the linear part to constant coefficients that was given in Ide et al. (2002) and show how the SVD of $\mathbf{Y}(t, t_0)$ and the finite time Lyapunov exponents are used in the construction of the transformation.

The constant matrix $\mathbf{D}$ in Eq. (15) will be specified in relation to the dynamics of $\mathbf{y} = \mathbf{A}(t)\mathbf{y}$. In order to do this we will relate the fundamental solution matrices $\mathbf{Y}(t, t_0)$ and $\mathbf{W}(t, t_0)$ corresponding to the evolution of trajectories of the purely linear homogeneous parts of Eqs. (8) and (15). Recall that the fundamental solution matrices satisfy:

$$\mathbf{Y}(t, t_0) = \mathbf{A}(t)\mathbf{Y}(t, t_0), \tag{A3}$$

$$\mathbf{W}(t, t_0) = \mathbf{D}\mathbf{W}(t, t_0), \tag{A4}$$

and since $\mathbf{D}$ will be chosen to be a constant matrix, it follows that:

$$\mathbf{W}(t, t_0) = e^{(t-t_0)\mathbf{D}}. \tag{A5}$$

Since evolution must commute with the change of coordinates, as shown in the diagram (A6),

$$
\begin{array}{ccc}
\boldsymbol{y}(t_0) & \xrightarrow{\;\mathbf{T}(t_0)\;} & \boldsymbol{w}(t_0) \\
\mathbf{Y}(t,t_0)\downarrow & & \downarrow \mathbf{W}(t,t_0) \\
\boldsymbol{y}(t) & \xrightarrow[\;\mathbf{T}(t)\;]{} & \boldsymbol{w}(t)
\end{array} \tag{A6}
$$

we have

$$
\begin{aligned}
\mathbf{Y}(t, t_0) &= \mathbf{T}^{-1}(t)\mathbf{W}(t, t_0)\mathbf{T}(t_0) \\
&= \mathbf{T}^{-1}(t)e^{\mathbf{D}(t-t_0)}\mathbf{T}(t_0)
\end{aligned} \tag{A7}
$$

Evaluating this expression at $t = t_L$, and using the SVD representation for $\mathbf{Y}(t, t_0)$, gives:

$$
\begin{aligned}
\mathbf{Y}(t_L, t_0) &= \mathbf{T}^{-1}(t_L)\mathbf{W}(t_L, t_0)\mathbf{T}(t_0) \\
&= \mathbf{T}^{-1}(t_L)e^{(t_L-t_0)\mathbf{D}}\mathbf{T}(t_0) \\
&= \mathbf{B}(t_L, t_0)e^{\Sigma(t_L, t_0)}\mathbf{R}^T(t_L, t_0),
\end{aligned} \tag{A8}
$$

from which it follows that:

$$
\mathbf{B}^T(t_L, t_0)\mathbf{T}^{-1}(t_L)e^{(t_L-t_0)\mathbf{D}}
$$
$$
\mathbf{T}(t_0)\mathbf{R}(t_L, t_0)e^{-\Sigma(t_L, t_0)} = \mathbf{id}. \tag{A9}
$$

This equality is satisfied with the choices:

$$\mathbf{D} = \Sigma(t_L, t_0)/(t_L - t_0), \tag{A10}$$

$$\mathbf{T}(t_0) = \mathbf{R}^T(t_L, t_0), \tag{A11}$$

$$\mathbf{T}(t_L) = \mathbf{B}^T(t_L, t_0). \tag{A12}$$

Therefore we see that $\mathbf{D}$ is a diagonal matrix whose eigenvalues are the finite time Lyapunov exponents.

Finally, from Eq. (A7), the SVD expression for $\mathbf{Y}(t, t_0)$, Eqs. (A10) and (A11) we have the transformation:

$$
\begin{aligned}
\mathbf{T}(t) &= e^{(t-t_0)\mathbf{D}}\mathbf{T}(t_0)\mathbf{Y}^{-1}(t, t_0) \\
&= e^{(t-t_0)\mathbf{D}}\mathbf{R}^T(t_L, t_0) \\
&\qquad \mathbf{R}(t, t_0)e^{-\Sigma(t, t_0)}\mathbf{B}^T(t, t_0).
\end{aligned} \tag{A13}
$$

So the transformation to constant coefficients is expressed entirely in terms of the SVD of $\mathbf{Y}(t, t_0)$. Furthermore, the use of the SVD representation allows the orthogonal matrices $\mathbf{R}(t, t_0)$ and $\mathbf{B}(t, t_0)$ in Eq. (A2) to be represented in terms of two scalar angle variables, while the exponential behaviour of $\mathbf{Y}(t, t_0)$ is incorporated into $\Sigma(t, t_0)$, which prevents the numerical difficulties which might occur from working directly with $\mathbf{Y}(t, t_0)$. The resulting system of differential equations is solved using a fourth-order Runge-Kutta scheme; full details of this procedure are given in Ide et al. (2002).

Edited by: P. H. Haynes
Reviewed by: two referees

References

Balasuriya, S.: Gradient evolution for potential vorticity flows, Non. Proc. Geophys., 8, 253–263, 2001.

Balasuriya, S. and Jones, C. K. R. T.: Diffusive draining and growth of eddies, Non. Proc. Geophys., 8, 241–251, 2001.

Behringer, R. P., Meyers, S. D., and Swinney, H. L.: Chaos and mixing in a geostrophic flow, Phys. Fluids A, 3, 1243–1249, 1991.

Bower, A. S.: A simple kinematic mechanism for mixing fluid parcels across a meandering jet, J. Phys. Oceanogr., 21, 173–180, 1991.

Coulliette, C. and Wiggins, S.: Intergyre transport in a wind-driven, quasigeostrophic double gyre: An application of lobe dynamics, Non. Proc. Geophys., 8, 69–94, 2001.

Dieci, L., Russell, R. D., and Vleck, E. S. V.: On the computation of Lyapunov exponents for continuous dynamical systems, SIAM J. Numer. Anal., 34(1), 402–423, 1997.

Dijkstra, H. A. and Katsman, C. A.: Temporal variability of the wind-driven quasi-geostrophic double gyre ocean circulation: basic bifurcation diagram, Geophys. Astrophys. Fluid Dynamics, 85, 195–232, 1997.

Dritschel, D.: Contour dynamics and contour surgery: Numerical algorithms for extended, high-resolution modelling of vortex dynamics in two-dimensional, inviscid, incompressible flows, Comp. Phys. Rep., 10, 77–146, 1989.

Dritschel, D. G. and Ambaum, M. H. P.: A contour-advective semi-Lagrangian numerical algorithm for simulating fine-scale conservative dynamical fields, Q. J. R. Meteorol. Soc., 123, 1097–1130, 1997.

Duan, J. and Wiggins, S.: Lagrangian transport and chaos in the near wake of the flow around an obstacle: a numerical implementation of lobe dynamics, Non. Proc. Geophys., 4, 125–136, 1997.

Duan, J. Q. and Wiggins, S.: Fluid exchange across a meandering jet with quasi-periodic time variability, J. Phys. Oceanogr., 26, 1176–1188, 1996.

Dutkiewicz, S., Griffa, A., and Olson, D. B.: Particle diffusion in a meandering jet, J. Geophys. Res., 98, 16 487–16 500, 1993.

Farrell, B. F. and Ioannou, P. J.: Generalized stability theory. part I: Autonomous operators, J. Atmos. Sci., 53, 2025–2040, 1996.

Geist, K., Parlitz, U., and Lauterborn, W.: Comparison of different methods for computing Lyapunov exponents, Prog. Theor. Phys., 83, 875–893, 1990.

Greene, J. M. and Kim, J.-S.: The calculation of Lyapunov spectra, Physica D, 24, 213–225, 1987.

Haller, G.: Finding finite-time invariant manifolds in two-dimensional velocity fields, Chaos, 10, 99–108, 2000.

Haller, G. and Poje, A.: Finite time transport in aperiodic flows, Physica D, 119, 352–380, 1998.

Haller, G. and Yuan, G.: Lagrangian coherent structures and mixing in two-dimensional turbulence, Physica D, 147, 352–370, 2000.

Ide, K., Small, D., and Wiggins, S.: Distinguished hyperbolic trajectories in time dependent fluid flows: analytical and computational approach for velocity fields defined as data sets, Non. Proc. Geophys. , 9, 237–263, 2002.

Ju, N., Small, D., and Wiggins, S.: Existence and computation of hyperbolic trajectories of aperiodically time-dependent vector fields and their approximations, Int. J. Bif. Chaos, 13, 1449–1457, 2003.

Kirwan, A. D., Toner, M., and Kantha, L.: Predictability, uncertainty, and hyperbolicity in the ocean, Int. J. Eng. Sci., 41, 249–258, 2003.

Kuznetsov, L., Toner, M., Kirwan, A. D., Jones, C. K. R. T., Kantha, L. H., and Choi, J.: The loop current and adjacent rings delineated by Lagrangian analysis of the near-surface flow, J. Marine Res., 60, 405–429, 2002.

Lapeyre, G.: Characterization of finite-time Lyapunov exponents and vectors in two-dimensional turbulence, Chaos, 12, 688–698, 2002.

Legras, B. and Vautard, R.: A guide to Liapunov vectors, in Proceedings of the 1995 ECMWF Seminar on Predictability, edited by Palmer, T., 143–156, 1996.

Lozier, M. S. and Riser, S.: Potential vorticity sources and sinks in a quasi-geostrophic ocean: beyond western boundary currents, J. Phys. Oceanogr., 20, 1608–1627, 1989.

Malhotra, N. and Wiggins, S.: Geometric structures, lobe dynamics, and Lagrangian transport in flows with aperiodic time-dependence, with applications to Rossby wave flow, J. Nonlinear Science, 8, 401–456, 1998.

Mancho, A., Small, D., Wiggins, S., and Ide, K.: Computation of stable and unstable manifolds of hyperbolic trajectories in two-dimensional, aperiodically time-dependent vector fields, Physica D, 182, 188–222, 2003.

Meyers, S. D.: Cross-frontal mixing in a meandering jet, J. Phys. Oceanogr., 24, 1641–1646, 1994.

Miller, P. D., Jones, C. K. R. T., Rogerson, A. M., and Pratt, L. J.: Quantifying transport in numerically generated velocity fields, Physica D, 110, 105–122, 1997.

Miller, P. D., Pratt, L. J., Helfrich, K. R., and Jones, C. K. R. T.: Chaotic transport of mass and potential vorticity for an island recirculation, J. Phys. Oceanogr., 32, 80–102, 2002.

Ottino, J.: The Kinematics of Mixing: Stretching, Chaos, and Transport, Cambridge University Press, Cambridge, 1989.

Poje, A. C. and Haller, G.: Geometry of cross-stream mixing in a double-gyre ocean model, J. Phys. Oceanogr., 29, 1649–1665, 1999.

Poje, A. C., Toner, M., Kirwan, A. D., and Jones, C. K. R. T.: Drifter launch strategies based on Lagrangian templates, J. Phys. Oceanogr., 32, 1855–1869, 2002.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P.: Numerical Recipes in C, Cambridge University Press, 1992.

Rogerson, A. M., Miller, P. D., Pratt, L. J., and Jones, C. K. R. T.: Lagrangian motion and fluid exchange in a barotropic meandering jet, J. Phys. Oceanogr., 29, 2635–2655, 1999.

Rowley, C.: A Modeling Study of the North Atlantic Current, Ph.D. thesis, University of Rhode Island, Kingston, RI, 1996.

Samelson, R. M.: Fluid exchange across a meandering jet, J. Phys. Oceanogr., 22, 431–440, 1992.

Waseda, T. and Mitsudera, H.: Chaotic advection of the shallow Kuroshio coastal waters, J. Oceanogr., 58, 627–638, 2002.

Wiggins, S.: Chaotic Transport in Dynamical Systems, Springer, New York, 1992.

Yuan, G.-C., Pratt, L. J., and Jones, C. K. R. T.: Barrier destruction and Lagrangian predictabilty at depth in a meandering jet, Dyn. Atmos. Oceans, 35, 41–61, 2001.