

---

---

## **Abstract**



---

---

## **Sammendrag**



---

---

## Preface



---

---

## Table of Contents

<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>Notation</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Theory</b>	<b>3</b>
2.1 Solving systems of ordinary differential equations . . . . .	3
2.1.1 The Runge-Kutta family of numerical ODE solvers . . . . .	4
2.2 The type of flow systems considered . . . . .	6
2.3 Definition of Lagrangian coherent structures for three-dimensional flows . .	10
2.3.1 Hyperbolic LCSs . . . . .	10
<b>3 Method</b>	<b>13</b>
3.1 Arnold-Beltrami-Childress flow . . . . .	13
3.1.1 (Normal) vector field for a sinusoidal surface . . . . .	14
3.2 The method of geodesic level sets . . . . .	14
<b>4 Results</b>	<b>17</b>
<b>5 Discussion</b>	<b>25</b>
<b>6 Conclusions</b>	<b>27</b>
<b>References</b>	<b>29</b>
<b>A Appendix A</b>	<b>31</b>



---

---

## List of Figures

2.1	Geometric interpretation of the eigenvectors of the Cauchy-Green strain tensor	9
2.2	Aviici is love, Aviici is life . . . . .	9
3.1	Geometric interpretation of the eigenvectors of the Cauchy-Green strain tensor	14
3.2	Geometric interpretation of the eigenvectors of the Cauchy-Green strain tensor	15
4.1	Veni, vidi, Aviici . . . . .	17
4.2	Veni, vidi, Aviici . . . . .	18
4.3	Veni, vidi, Aviici . . . . .	18
4.4	Veni, vidi, Aviici . . . . .	19
4.5	Veni, vidi, Aviici . . . . .	19
4.6	Aviici is love, Aviici is life . . . . .	19
4.7	Aviici is love, Aviici is life . . . . .	20
4.8	Veni, vidi, Aviici . . . . .	20
4.9	Aviici is love, Aviici is life . . . . .	21
4.10	Aviici is love, Aviici is life . . . . .	21
4.11	Aviici is love, Aviici is life . . . . .	22
4.12	Aviici is love, Aviici is life . . . . .	22
4.13	Aviici is love, Aviici is life . . . . .	23
4.14	Aviici is love, Aviici is life . . . . .	23
4.15	Aviici is love, Aviici is life . . . . .	24



---

---

## List of Tables

2.1	Butcher tableau representing a generic $s$ -stage Runge-Kutta method . . . . .	6
2.2	Butcher tableau representation of a generic, embedded, explicit Runge-Kutta method . . . . .	7



---

---

## Notation

Newton's notation is used for differentiation with respect to time, i.e.:

$$\dot{f}(t) \equiv \frac{df(t)}{dt}.$$

Vectors are denoted by lowercase, upright, bold letters, like this:

$$\xi = (\xi_1, \xi_2, \dots, \xi_n).$$

The Euclidean norm of a vector  $\xi \in \mathbb{R}^n$  is denoted by:

$$\|\xi\| = \sqrt{\xi_1^2 + \xi_2^2 + \dots + \xi_n^2}.$$

Matrices and matrix representations of rank-2 tensors are denoted by uppercase, upright, bold letters, as follows:

$$\mathbf{A} = (a_{i,j}) = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}.$$



---

---

## **1 Introduction**

---



---

## 2 Theory

---

### 2.1 SOLVING SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS

In physics, like other sciences, modeling a system often equates to solving an initial value problem. An initial value problem can be described in terms of an ordinary differential equation (hereafter abbreviated to ODE) of the form

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \quad (2.1)$$

where  $x$  is an unknown function (scalar or vector) of time  $t$ . The function  $f$  is defined on an open subset  $\Omega$  of  $\mathbb{R} \times \mathbb{R}^n$ , where  $n$  is the number of spatial dimensions; that is, the number of components of  $x$ . The initial condition  $(t_0, x_0)$  is a point in the domain of  $f$ , i.e.,  $(t_0, x_0) \in \Omega$ . In higher dimensions (namely,  $n > 1$ ), the differential equation (2.1) generally extends to a coupled family of ODEs

$$\dot{x}_i(t) = f_i(t, x_1(t), x_2(t), \dots, x_n(t)), \quad x_i(t_0) = x_{i,0}, \quad i = 1, \dots, n. \quad (2.2)$$

The system is nonlinear if the function  $f$  in equation (2.1), or, if at least one of the functions  $\{f_i\}$  in equation (2.2), is nonlinear in one or more of its arguments. For the sake of notational simplicity, the discussion to follow in the rest of this section is based on the one-dimensional case, that is, system (2.1), for  $n = 1$ . However, all of the considerations also hold for  $n > 1$ .

Say that the solution of system (2.1) is sought at some time  $t_f$ . In order to approximate said solution numerically, the time variable must be discretized first. This is frequently done by defining

$$t_j = t_0 + j \cdot h, \quad (2.3)$$

where  $t_j$  is the time level  $j$  for integer  $j$ , and  $h$  is some increment which is smaller than  $t_f - t_0$ . Typically, the time increment is chosen such that an integer number of step lengths  $h$  equals the difference  $t_f - t_0$ . With the discretized time, the numerical solution of system (2.1) is found by successive applications of some numerical integration method. The Runge-Kutta family of numerical methods for ODE systems is a common choice, and will be elaborated upon in greater detail in section 2.1.1.

All numerical integration schemes fall into one of two categories; explicit and implicit methods. Explicit methods are characterized by computing the state of the system at a later time, based on the state of the system at the current time (in some cases, the state at earlier times are also considered). Implicit methods, however, involve the solution of an equation in which both the current and the later state of the system are involved. Thus, a generic, explicit method for computing the state of the system at time  $t + h$ , given its state at  $t$ , can be expressed as

$$x(t + h) = F(x(t)), \quad (2.4a)$$

while, for implicit methods, an equation of the sort

$$G(x(t), x(t + h)) = 0, \quad (2.4b)$$

is solved to find  $x(t + h)$ .

In general, implicit methods require the solution of a linear system at every time step. Clearly, implicit methods are more computationally demanding than explicit methods. The main selling point of implicit methods is that they are more numerically stable than explicit methods. This property means that implicit methods are particularly well-suited for *stiff* systems, i.e., physical systems with highly disparate time scales (Hairer and Wanner 1996, p.2). For such systems, most explicit methods are unstable, unless the time step  $h$  is made exceptionally small, rendering these methods practically useless. For *nonstiff* systems, however, implicit methods behave similarly to their explicit analogues in terms of numerical accuracy and convergence properties.

Irrespective of which numerical integration method is employed, one obtains an approximation of the true solution of the system (2.1) *at* the discrete time levels, that is,

$$x_j \approx x(t_j), \quad (2.5)$$

where  $x(t)$  is the exact solution at time  $t$ . The accuracy of the approximation, however, depends on both the numerical integration method and the time step length  $h$  used for the temporal discretization. One way of obtaining approximations of the true solution *inbetween* the discrete time levels is by means of interpolation — a numerical technique which will be elaborated upon in **SETT IN REF**. For nonlinear systems, analytical solutions usually do not exist. Thus, such systems are often analyzed by means of numerical methods.

### 2.1.1 The Runge-Kutta family of numerical ODE solvers

In numerical analysis, the Runge-Kutta family of methods is a popular collection of implicit and explicit iterative methods, used in temporal discretization in order to obtain numerical approximations of the *true* solutions of systems like (2.1). The German mathematicians C. Runge and M.W. Kutta developed the first of the family's methods at the turn of the twentieth century (Hairer, Nørsett, and Wanner 1993, p.134). The general outline of what is now known as a Runge-Kutta method is as follows:

#### **Definition 1** (Runge-Kutta methods).

Let  $s$  be an integer and  $\{a_{i,j}\}_{i,j=1}^s$ ,  $\{b_i\}_{i=1}^s$  and  $\{c_i\}_{i=1}^s$  be real coefficients.

Let  $h$  be the numerical step length used in the temporal discretization.

Then, the method

$$\begin{aligned} k_i &= f\left(t_n + c_i h, x_n + h \sum_{j=1}^s a_{i,j} k_j\right), \quad i = 1, \dots, s, \\ x_{n+1} &= x_n + h \sum_{i=1}^s b_i k_i, \end{aligned} \quad (2.6)$$

is called an *s-stage Runge-Kutta method* for the system (2.1).

The main reason to include multiple stages in a Runge-Kutta method is to improve the numerical accuracy of the computed solutions. The *order* of a Runge-Kutta method can be defined as follows:

**Definition 2** (*Order of Runge-Kutta methods*).

A Runge-Kutta method, given by equation (2.6), is of *order p* if, for sufficiently smooth systems (2.1), the local error  $e_n$  scales as  $h^{p+1}$ . That is:

$$e_n = \|x_n - u_{n-1}(t_n)\| \leq K h^{p+1} \quad (2.7)$$

where  $u_{n-1}(t)$  is the exact solution of the ODE in system (2.1) at time  $t$ , subject to the initial condition  $u_{n-1}(t_{n-1}) = x_{n-1}$ , and  $K$  is a numerical constant. This is true, if the Taylor series for the exact solution  $u_{n-1}(t_n)$  and the numerical solution  $x_n$  coincide up to (and including) the term  $h^p$ .

The *global* error

$$E_n = x_n - x(t_n), \quad (2.8)$$

where  $x(t)$  is the exact solution of system (2.1) at time  $t$ , accumulated by  $n$  repeated applications of the numerical method, can be estimated by

$$|E_n| \leq C \sum_{l=1}^n |e_l|, \quad (2.9)$$

where  $C$  is a numerical constant, depending on both the right hand side of the ODE in system (2.1) and the difference  $t_n - t_0$ . Making use of definition 2, the global error is limited from above by

$$\begin{aligned} |E_n| &\leq C \sum_{l=1}^n |e_l| \leq C \sum_{l=1}^n |K_l| h^{p+1} \leq C \max_l \{|K_l|\} n h^{p+1} \\ &\leq C \max_l \{|K_l|\} \frac{t_n - t_0}{h} h^{p+1} \leq \tilde{K} h^p, \end{aligned} \quad (2.10)$$

where  $\tilde{K}$  is a numerical constant. Equation (2.10) demonstrates that, for a  $p$ -th order Runge-Kutta method, the global error can be expected to scale as  $h^p$ .

In definition 1, the matrix  $(a_{i,j})$  is commonly called the *Runge-Kutta matrix*, while the coefficients  $\{b_i\}$  and  $\{c_i\}$  are known as the *weights* and *nodes*, respectively. Since the 1960s, it has been customary to represent Runge-Kutta methods, given by equation (2.6), symbolically, by means of mnemonic devices known as Butcher tableaus (Hairer, Nørsett, and Wanner 1993, p.134). The Butcher tableau for a general  $s$ -stage Runge-Kutta method, as introduced in definition 1, is presented in table 2.1. For explicit Runge-Kutta methods, the Runge-Kutta matrix  $(a_{i,j})$  is lower triangular. Similarly, for fully implicit Runge-Kutta methods, the Runge-Kutta matrix is upper triangular. The difference between explicit and implicit methods is outlined in equation (2.4).

During the first half of the twentieth century, a substantial amount of research was conducted in order to develop numerically robust, high-order, explicit Runge-Kutta methods. The idea

**Table 2.1:** Butcher tableau representing a generic  $s$ -stage Runge-Kutta method.

$c_1$	$a_{1,1}$	$a_{1,2}$	$\dots$	$a_{1,s}$
$c_2$	$a_{2,1}$	$a_{2,2}$	$\dots$	$a_{2,s}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$c_s$	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s}$
	$b_1$	$b_2$	$\dots$	$b_s$

was that using such methods would mean one could resort to larger time increments  $h$  without sacrificing precision in the computed solution. However, the required number of stages  $s$  grows quicker than linearly as a function of the required order  $p$ . It has been proven that, for  $p \geq 5$ , no explicit Runge-Kutta method of order  $p$  with  $s = p$  stages exists (Hairer, Nørsett, and Wanner 1993, p.173). This is one of the reasons for the attention shift from the latter half of the 1950s and onwards, towards so-called *embedded* Runge-Kutta methods.

The basic idea of embedded Runge-Kutta methods is that they, aside from the numerical approximation  $x_{n+1}$ , yield a second approximation  $\widehat{x}_{n+1}$ . The difference between the two approximations then provides an estimate of the local error of the less precise result, which can be used for automatic step size control (Hairer, Nørsett, and Wanner 1993, pp.167–168). The trick is to construct two independent, explicit Runge-Kutta methods which both use the *same* function evaluations. This results in practically obtaining the two solutions for the price of one, in terms of computational complexity. The Butcher tableau of a generic, embedded, explicit Runge-Kutta method is illustrated in table 2.2.

For embedded methods, the coefficients are tuned such that

$$x_{n+1} = x_n + h \sum_{i=1}^s b_i k_i \quad (2.11a)$$

is of order  $p$ , and

$$\widehat{x}_{n+1} = x_n + h \sum_{i=1}^s \widehat{b}_i k_i \quad (2.11b)$$

is of order  $\widehat{p}$ , typically with  $\widehat{p} = p + 1$ . Which of the solutions is used to continue the numerical integration, depends on the integration method in question. In the following, the solution which is *not* used to continue the integration, will be referred to as the *interpolant* solution.

## 2.2 THE TYPE OF FLOW SYSTEMS CONSIDERED

We consider flow in three-dimensional dynamical systems of the form

$$\dot{\mathbf{x}} = \mathbf{v}(t, \mathbf{x}), \quad \mathbf{x} \in \mathcal{U}, \quad t \in [t_0, t_1], \quad (2.12)$$

**Table 2.2:** Butcher tableau representation a generic, embedded, explicit Runge-Kutta method.

0					
$c_2$	$a_{2,1}$				
$c_3$	$a_{3,1}$	$a_{3,2}$			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
$c_s$	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s-1}$	
		$b_1$	$b_2$	$\dots$	$b_{s-1}$
		$\widehat{b}_1$	$\widehat{b}_2$	$\dots$	$\widehat{b}_{s-1}$
		$\widehat{b}_s$			

i.e., systems defined for the finite time interval  $[t_0, t_1]$  on an open, bounded subset  $\mathcal{U}$  of  $\mathbb{R}^3$ . In addition, the velocity field  $\mathbf{v}$  is assumed to be smooth in its arguments. Depending on the exact nature of the velocity field  $\mathbf{v}$ , analytical particle trajectories, that is, analytical solutions of system (2.12), may or may not exist. The flow particles are assumed to be infinitesimal and massless, i.e., non-interacting *tracers* of the overall circulation.

Letting  $\mathbf{x}(t; t_0, \mathbf{x}_0)$  denote the trajectory of a tracer in the system given by equation (2.12), the flow map is defined as

$$\Phi_{t_0}^t(\mathbf{x}_0) = \mathbf{x}(t; t_0, \mathbf{x}_0), \quad (2.13)$$

hence, the flow map describes the movement of tracers from one point in time to another mathematically. In general, the flow map is as smooth as the underlying velocity field (cf. system (2.12)) (Farazmand and Haller 2012a). In Lagrangian flow analysis, the *Jacobian matrix* of the flow map  $\Phi_{t_0}^t$  plays a significant role. Component-wise, the Jacobian matrix of a general vector-valued function  $\mathbf{f}$  is defined as

$$(\nabla \mathbf{f})_{i,j} = \frac{\partial f_i}{\partial x_j}, \quad \mathbf{f} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots), \quad (2.14)$$

which, for our three-dimensional flow, reduces to

$$\nabla \mathbf{f} = \begin{pmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{pmatrix}. \quad (2.15)$$

Making use of the definition of the flow map (cf. equation (2.13)) in conjunction with equation (2.12), one finds the following ordinary differential equation which describes the time evolution of the flow map:

$$\dot{\Phi} = \mathbf{v}(t, \Phi), \quad (2.16)$$

where  $t_0$ ,  $t$  and  $\mathbf{x}_0$  have been omitted in order to avoid notational clutter. These are, however, implicit by context. As the nabla operator is time-independent, equation (2.16) immediately

yields an ordinary differential equation for the time development of the directional derivative of the flow map, namely

$$\frac{d}{dt}(\hat{\mathbf{u}} \cdot \nabla)\phi = (\hat{\mathbf{u}} \cdot \nabla)v(t, \phi), \quad (2.17)$$

which holds along any constant unit vector  $\hat{\mathbf{u}}$ . On a regular Cartesian grid, equation (2.17) provides a coupled set of ordinary differential equations describing the time evolution of each component of the Jacobian of the flow map:

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial\phi_i}{\partial x_j}\right) &= \sum_k \frac{\partial v_i}{\partial x_k}\Big|_{(t,\phi)} \frac{\partial\phi_k}{\partial x_j}\Big|_t, \\ \frac{\partial\phi_i}{\partial x_j}\Big|_{t_0} &= \delta_{ij}, \quad \mathbf{x}_0 \in \mathcal{U}, \quad t \in [t_0, t_1], \end{aligned} \quad (2.18)$$

where the Kronecker delta is defined as

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \quad (2.19)$$

The initial conditions for the Jacobi components reflect the fact that, for a regular Cartesian grid, the directional derivative of the  $x$  coordinate in the  $x$  direction is  $q$ , but zero in the  $y$  and  $z$  directions.

For sufficiently smooth velocity fields, the flow map Jacobian  $\nabla\Phi_{t_0}^t$  can be computed, which allows for the right Cauchy-Green strain tensor field to be defined as

$$\mathbf{C}_{t_0}^t(\mathbf{x}_0) = (\nabla\Phi_{t_0}^t(\mathbf{x}_0))^* (\nabla\Phi_{t_0}^t(\mathbf{x}_0)), \quad (2.20)$$

where the asterisk refers to the adjoint operation, which, because the Jacobian  $\nabla\Phi_{t_0}^t$  is real-valued, equates to matrix transposition. Moreover, as the Jacobian of the flow map is invertible, the Cauchy-Green strain tensor  $\mathbf{C}_{t_0}^t(\mathbf{x}_0)$  is symmetric and positive definite (Farazmand and Haller 2012a). Thus, it has three real, positive eigenvalues and orthogonal, real eigenvectors. Its eigenvalues  $\lambda_i$  and corresponding unit eigenvectors  $\xi_i$  are defined by

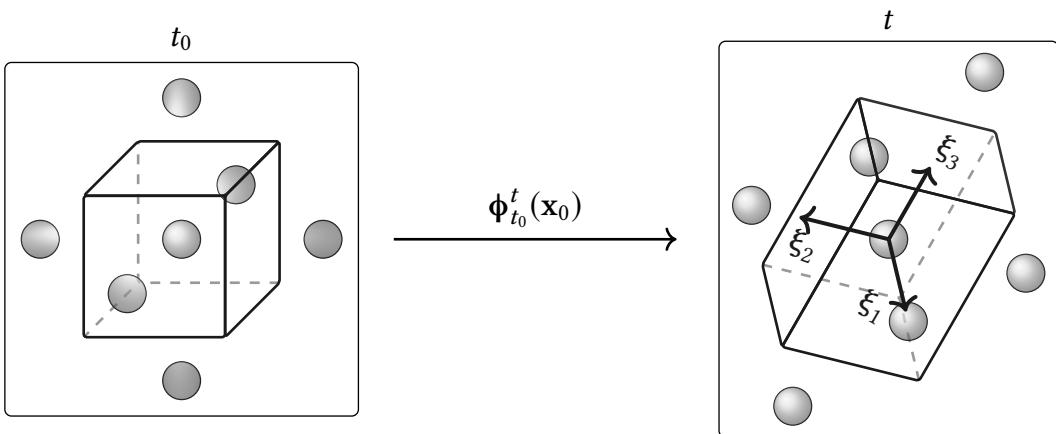
$$\begin{aligned} \mathbf{C}_{t_0}^t(\mathbf{x}_0)\xi_i(\mathbf{x}_0) &= \lambda_i\xi_i(\mathbf{x}_0), \quad i = 1, 2, 3, \\ \langle \xi_i(\mathbf{x}_0), \xi_j(\mathbf{x}_0) \rangle &= \delta_{ij}, \quad 0 < \lambda_1(\mathbf{x}_0) \leq \lambda_2(\mathbf{x}_0) \leq \lambda_3(\mathbf{x}_0), \end{aligned} \quad (2.21)$$

where the Kronecker delta is defined in equation (2.19), and the dependence of  $\lambda_i$  and  $\xi_i$  on  $t_0$  and  $t$  has been suppressed, for the sake of notational transparency. The geometric interpretation of equation (2.21) is that a fluid element undergoes the most stretching along the  $\xi_3$  axis, less stretching along the  $\xi_2$  axis, and the least stretching along the  $\xi_1$  axis. This concept is shown in figure 2.1.

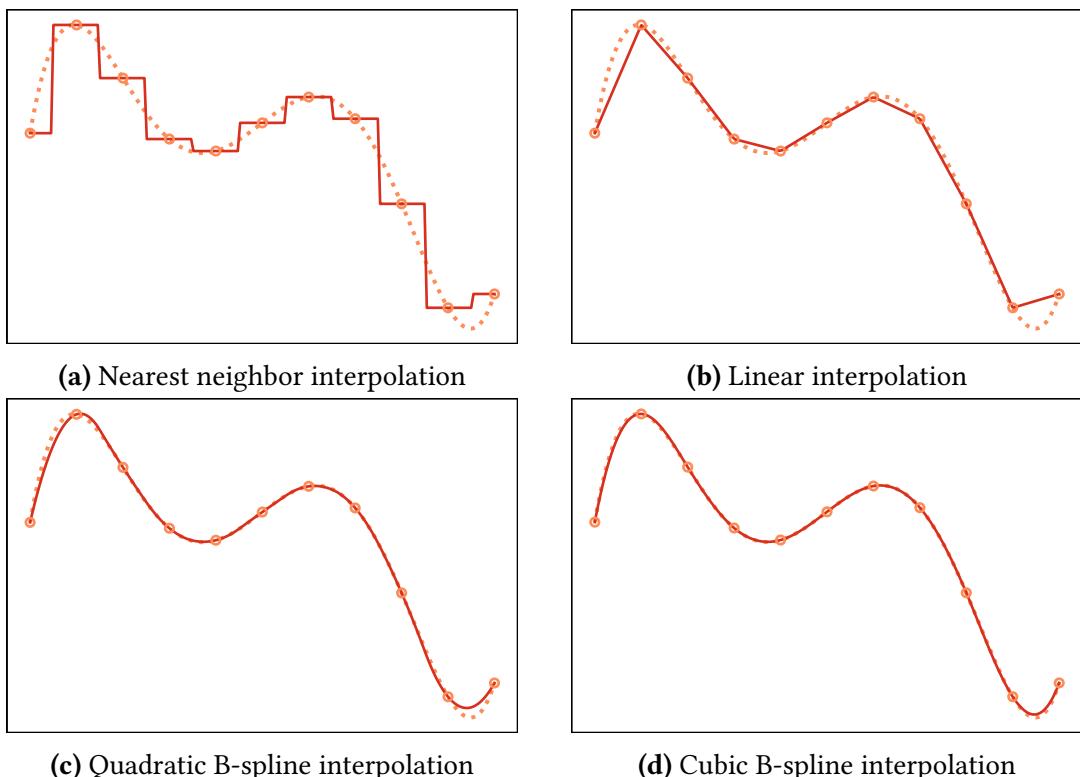
As the stretch factors along the  $\xi_i$  axes are given by the square roots of the corresponding eigenvalues, for incompressible flow, the eigenvalues satisfy

$$\lambda_1(\mathbf{x}_0)\lambda_2(\mathbf{x}_0)\lambda_3(\mathbf{x}_0) = 1 \quad \forall \mathbf{x}_0 \in \mathcal{U}, \quad (2.22)$$

where, in the context of tracer advection, incompressibility is equivalent to the velocity field  $\mathbf{v}$  being divergence-free (i.e.,  $\nabla \cdot \mathbf{v} \equiv 0$  in system (2.12)).



**Figure 2.1:** Geometric interpretation of the eigenvectors of the Cauchy-Green strain tensor. The central unit cell is stretched and deformed under the flow map  $\phi_{t_0}^t(\mathbf{x}_0)$ . The local stretching is the largest in the direction of  $\xi_3$ , the eigenvector which corresponds to the largest eigenvalue,  $\lambda_3$ , of the Cauchy-Green strain tensor, defined in equation (2.21). Along the  $\xi_i$  axes, the stretch factors are given by  $\sqrt{\lambda_i}$ , respectively.



**Figure 2.2:** Spline interpolation of orders 0 through to 3 applied to a higher order polynomial (dashed). The sampling points are shown as hollow circles. Observe how higher order splines yield increasingly accurate and smooth interpolations.

## 2.3 DEFINITION OF LAGRANGIAN COHERENT STRUCTURES FOR THREE-DIMENSIONAL FLOWS

Lagrangian coherent structures (henceforth abbreviated to LCSs) can be described as time-evolving surfaces which shape coherent trajectory patterns in dynamical systems, defined over a finite time interval (Haller 2010). There are three main types of LCSs, namely *elliptic*, *hyperbolic* and *parabolic*. Roughly speaking, parabolic LCSs outline cores of jet-like trajectories, elliptic LCSs describe vortex boundaries, whereas hyperbolic LCSs are comprised of overall attractive or repelling manifolds. As such, hyperbolic LCSs practically act as organizing centers of observable tracer patterns (Onu, Huhn, and Haller 2015). Because hyperbolic LCSs provide the most readily applicable insight in terms of forecasting flow in e.g. oceanic currents, such structures have been the focus of this project.

### 2.3.1 Hyperbolic LCSs

The identification of LCSs for reliable forecasting requires sufficiency and necessity conditions, supported by mathematical theorems. Haller (2010) derived a variational LCS theory based on the Cauchy-Green strain tensor, defined by equation (2.20), from which the aforementioned conditions follow. The immediately relevant parts of Haller's theory are given in definitions 3–6 (Haller 2010).

**Definition 3** (*Normally repellent material surfaces*).

A *normally repellent material surface* over the time interval  $[t_0, t_0 + T]$  is a compact material surface segment  $\mathcal{M}(t)$  which is overall repelling, and on which the normal repulsion rate is greater than the tangential repulsion rate.

A *material surface* is a smooth surface  $\mathcal{M}(t_0)$  at time  $t_0$ , which is advected by the flow map, given by equation (2.13), into a dynamic material line  $\mathcal{M}(t) = \Phi_{t_0}^t(\mathcal{M}(t_0))$ . The required *compactness* of the material surface segment signifies that, in some sense, it must be topologically well-behaved. That the material surface is *overall repelling* means that nearby trajectories are repelled from, rather than attracted towards, the material surface. Lastly, requiring that the *normal* repulsion rate is greater than the *tangential* repulsion rate means that nearby trajectories are in fact driven away from the material surface, rather than being stretched along with it due to shear stress.

**Definition 4** (*Repelling LCS*).

A *repelling LCS* over the time interval  $[t_0, t_0 + T]$  is a normally repelling material surface  $\mathcal{M}(t_0)$  whose normal repulsion admits a pointwise non-degenerate maximum relative to any nearby material surface  $\widehat{\mathcal{M}}(t_0)$ .

**Definition 5** (*Attracting LCS*).

An *attracting LCS* over the time interval  $[t_0, t_0 + T]$  is defined as a repelling LCS over the *backward* time interval  $[t_0 + T, t_0]$ .

**Definition 6** (*Hyperbolic LCS*).

A *hyperbolic LCS* over the time interval  $[t_0, t_0 + T]$  is a *repelling* or *attracting* LCS over the *same* time interval.

Note that the above definitions associate LCSs with the time interval  $I$  over which the dynamical system under consideration is known, or, at the very least, where information regarding the behaviour of tracers, is sought. Generally, LCSs obtained over a time interval  $I$  do not necessarily exist over different time intervals (Farazmand and Haller 2012a).

For sufficiently smooth three-dimensional flow, the above definitions can be summarized as a set of mathematical existence criteria, based on the Cauchy-Green strain tensor (Haller 2010; Farazmand and Haller 2012a; Karrasch 2012; Farazmand and Haller 2012b). These are given in theorem 1.

**Theorem 1** (Sufficient and necessary conditions for LCSs in three-dimensional flows). *Consider a compact material surface  $\mathcal{M}(t) \subset \mathcal{U}$  evolving over the time interval  $[t_0, t_0 + T]$ . Then  $\mathcal{M}(t)$  is a repelling LCS over  $[t_0, t_0 + T]$  if and only if all of the following holds for all initial conditions  $\mathbf{x}_0 \in \mathcal{M}(t_0)$ :*

$$\lambda_2(\mathbf{x}_0) \neq \lambda_3(\mathbf{x}_0) > 1, \quad (2.23a)$$

$$\langle \xi_3(\mathbf{x}_0), \mathbf{H}_{\lambda_3}(\mathbf{x}_0) \xi_3(\mathbf{x}_0) \rangle < 0 \quad (2.23b)$$

$$\xi_3(\mathbf{x}_0) \perp \mathcal{M}(t_0), \quad (2.23c)$$

$$\langle \nabla \lambda_3(\mathbf{x}_0), \xi_3(\mathbf{x}_0) \rangle = 0. \quad (2.23d)$$

In theorem 1,  $\langle \cdot, \cdot \rangle$  signifies the Euclidean inner product, and  $\mathbf{H}_{\lambda_3}$  denotes the Hessian matrix of the largest eigenvalues of the Cauchy-Green strain tensor field. Component-wise, the Hessian matrix of a general, smooth, scalar-valued function  $f$  is defined as

$$(\mathbf{H}_f)_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j}, \quad (2.24)$$

which, for our three-dimensional flow, reduces to

$$\mathbf{H}_f = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial x \partial z} \\ \frac{\partial^2 f}{\partial y \partial x} & \frac{\partial^2 f}{\partial y^2} & \frac{\partial^2 f}{\partial y \partial z} \\ \frac{\partial^2 f}{\partial z \partial x} & \frac{\partial^2 f}{\partial z \partial y} & \frac{\partial^2 f}{\partial z^2} \end{pmatrix}. \quad (2.25)$$

Condition (2.23a) ensures that the normal repulsion rate is larger than the tangential stretch due to shear strain along the LCS, in accordance with definition 3. Conditions (2.23c) and (2.23d) suffice to enforce that the normal repulsion rate attains a local extremum along the LCS, relative to all nearby material surfaces. Lastly, condition (2.23b) ensures that this is a strict local maximum.

Forslag til teori-outlines

- Interpolasjon
- Integrasjon
- Cauchy-Green
- Variasjonsligning
- LCS, spesifikt hyperbolske i 3D
- Hva er en mangfoldighet?
- Oettingers argument del 2

### 3 Method

In order to identify LCSs in three-dimensional flow by means of geodesic level set approximations, a system which has been studied extensively in the literature was chosen. The system is a simple example of a fluid flow which can exhibit chaotic behaviour (Frisch 1995, p.204).

#### 3.1 ARNOLD-BELTRAMI-CHILDRESS FLOW

The Arnold-Beltrami-Childress (hereafter abbreviated to ABC) flow is a three-dimensional incompressible velocity field which solves the Euler equations exactly. In terms of the Cartesian coordinate vector  $\mathbf{x} = (x, y, z)$ , the system can be expressed mathematically as

$$\dot{\mathbf{x}} = \mathbf{v}(t, \mathbf{x}) = \begin{pmatrix} A \sin(z) + C \cos(y) \\ B \sin(x) + A \cos(z) \\ C \sin(y) + B \cos(x) \end{pmatrix}, \quad (3.1)$$

where  $A$ ,  $B$  and  $C$  are spatially invariant parameters which dictate the nature of the flow pattern. The inherent periodicity with regards to the Cartesian coordinates naturally leads to a domain of interest  $\mathcal{U} = [0, 2\pi]^3$  with periodic boundary conditions imposed along all three Cartesian axes.

For stationary flow, the parameter values

$$A = \sqrt{3}, \quad B = \sqrt{2}, \quad C = 1 \quad (3.2)$$

were used, as has been common in the literature, for instance in the article by Oettinger and Haller (2016), as these values are known for exhibiting chaotic trajectories (Zhao et al. 1993).

Rundt her kan sikkert språket forbedres / brødtekst genereres. Lar det være et problem for fremtidige A.

In order to enforce chaotic behaviour, the parameters  $A$ ,  $B$  and  $C$  can be modified to be temporally aperiodic as follows:

$$\begin{aligned} A &= \sqrt{3}, \\ B &= \sqrt{2} [1 + k_0 \tanh(k_1 t) \cos((k_2 t)^2)], \\ C &= 1 + k_0 \tanh(k_1 t) \sin((k_3 t)^2), \end{aligned} \quad (3.3)$$

where the parameter values

$$k_0 = 0.3, \quad k_1 = 0.5, \quad k_2 = 1.5, \quad k_3 = 1.8 \quad (3.4)$$

were used, like in Oettinger and Haller (2016).

### 3.1.1 (Normal) vector field for a sinusoidal surface

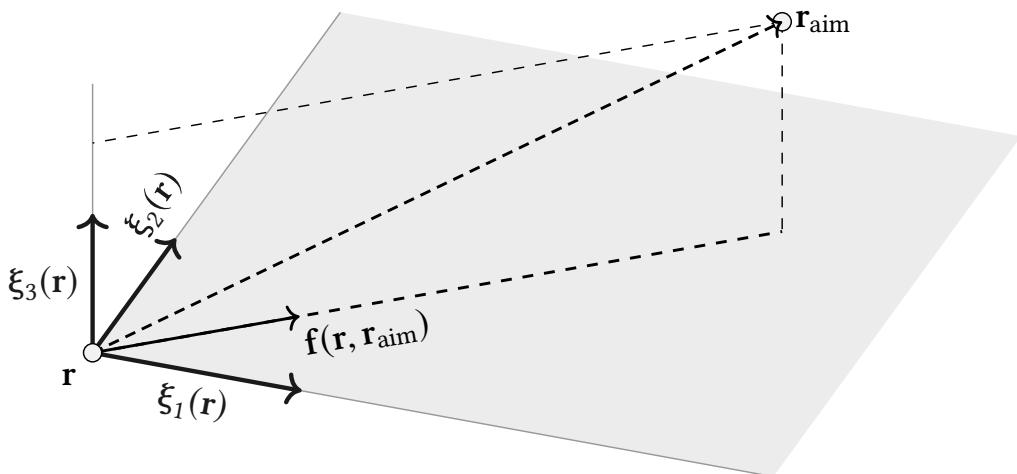
For the surface implicitly defined as the zeros of the function

$$f(\mathbf{r}) = A \sin(\omega_x x) \sin(\omega_y y) + (z_0 - z), \quad (3.5)$$

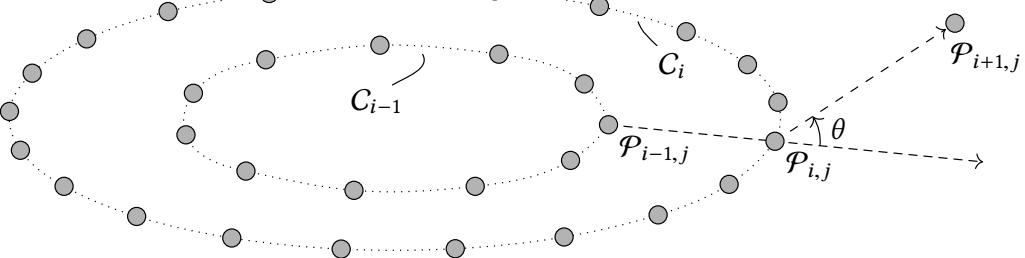
one possible choice for its normal vector field is

$$\mathbf{n}(\mathbf{r}) = \begin{pmatrix} A\omega_x \cos(\omega_x x) \sin(\omega_y y) \\ A\omega_y \sin(\omega_x x) \cos(\omega_y y) \\ -1 \end{pmatrix}. \quad (3.6)$$

## 3.2 THE METHOD OF GEODESIC LEVEL SETS



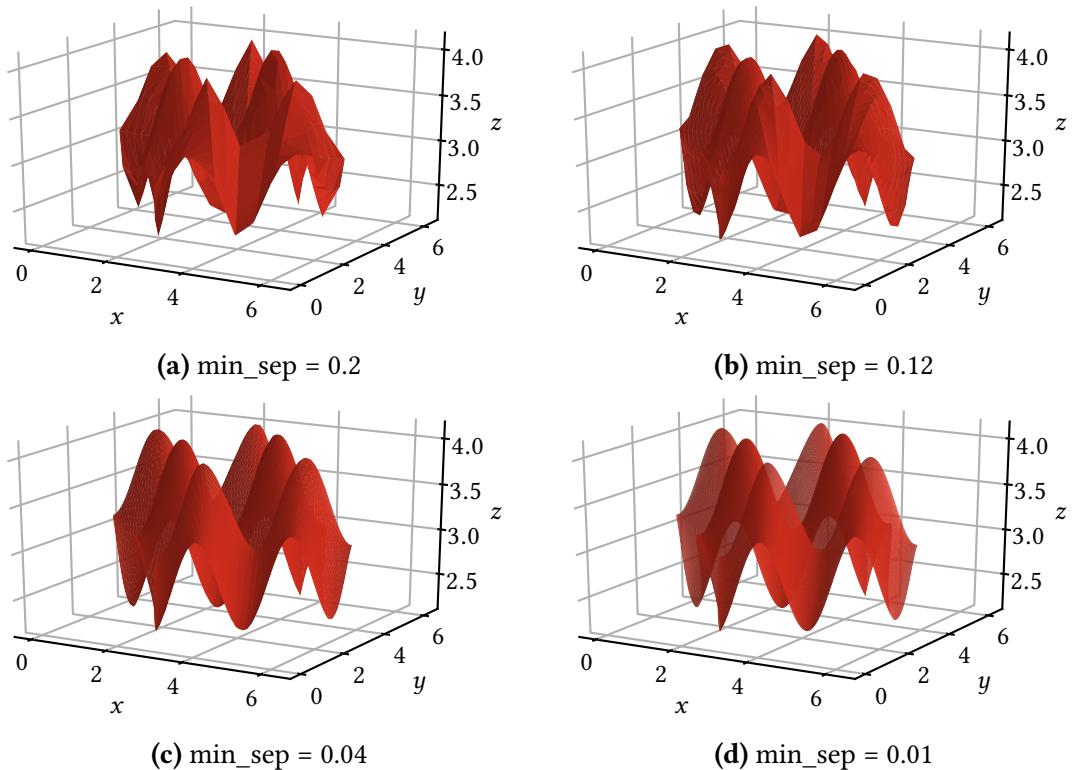
**Figure 3.1:** Geometric interpretation of the eigenvectors of the Cauchy-Green strain tensor. The central unit cell is stretched and deformed under the flow map  $\phi_{t_0}^t(\mathbf{x}_0)$ . The local stretching is the largest in the direction of  $\xi_3$ , the eigenvector which corresponds to the largest eigenvalue,  $\lambda_3$ , of the Cauchy-Green strain tensor, defined in equation (2.21). Along the  $\xi_i$  axes, the stretch factors are given by  $\sqrt{\lambda_i}$ , respectively.



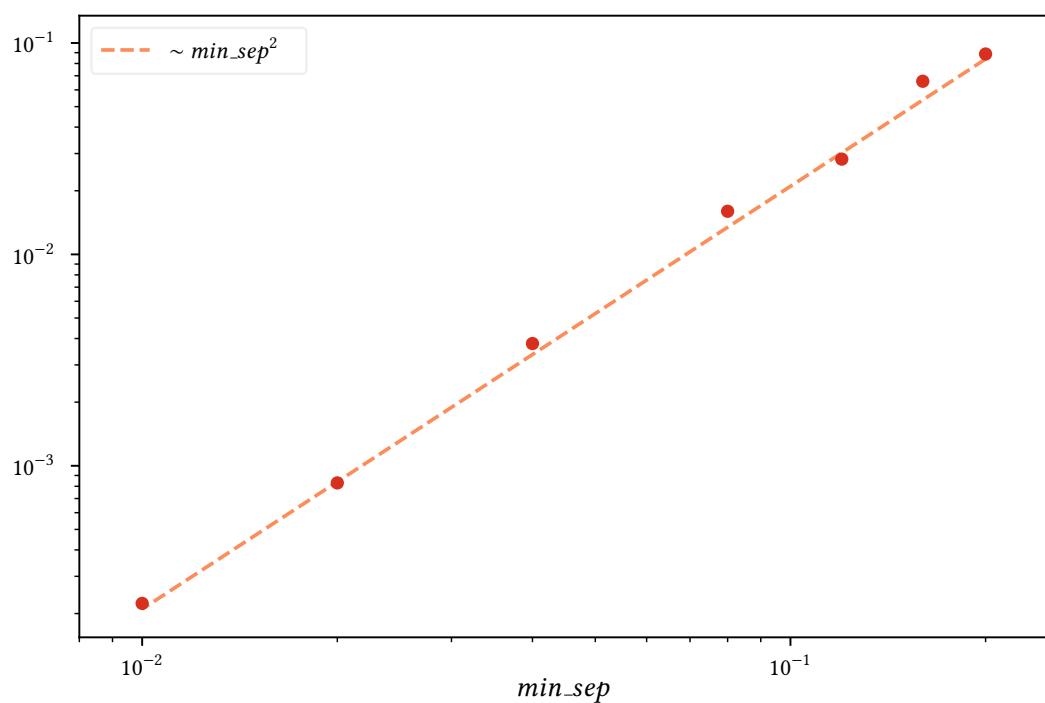
**Figure 3.2:** Geometrix interpretation of the eigenvectors of the Cauchy-Green strain tensor. The central unit cell is stretched and deformed under the flow map  $\Phi_{t_0}^t(\mathbf{x}_0)$ . The local stretching is the largest in the direction of  $\xi_3$ , the eigenvector which corresponds to the largest eigenvalue,  $\lambda_3$ , of the Cauchy-Green strain tensor, defined in equation (2.21). Along the  $\xi_i$  axes, the stretch factors are given by  $\sqrt{\lambda_i}$ , respectively.



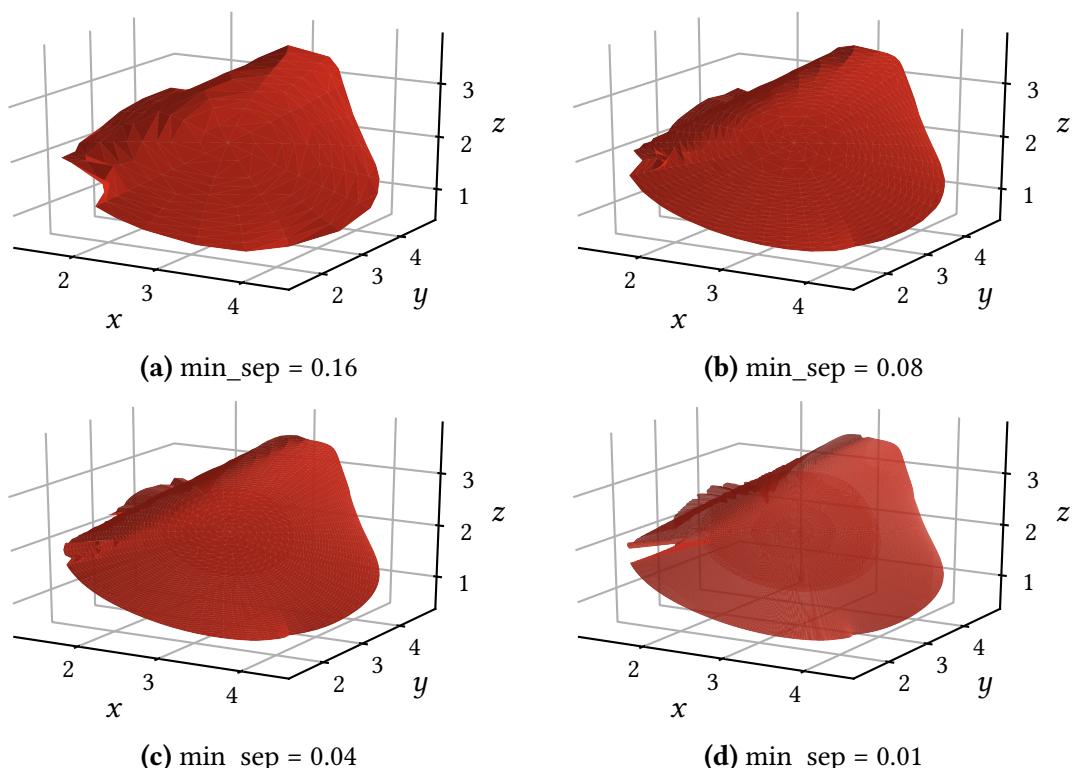
## 4 Results



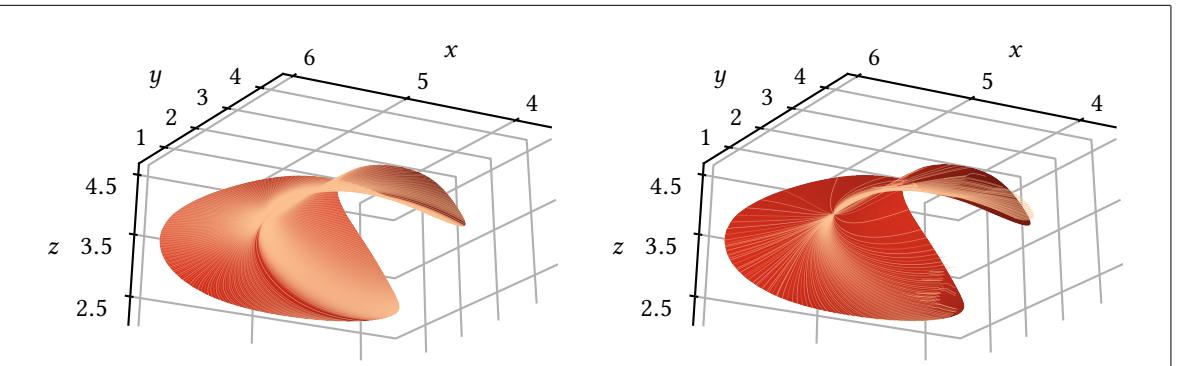
**Figure 4.1:** Manifolds generated from analytically determined vector field.



**Figure 4.2:** RMS error of sinusoidal manifolds

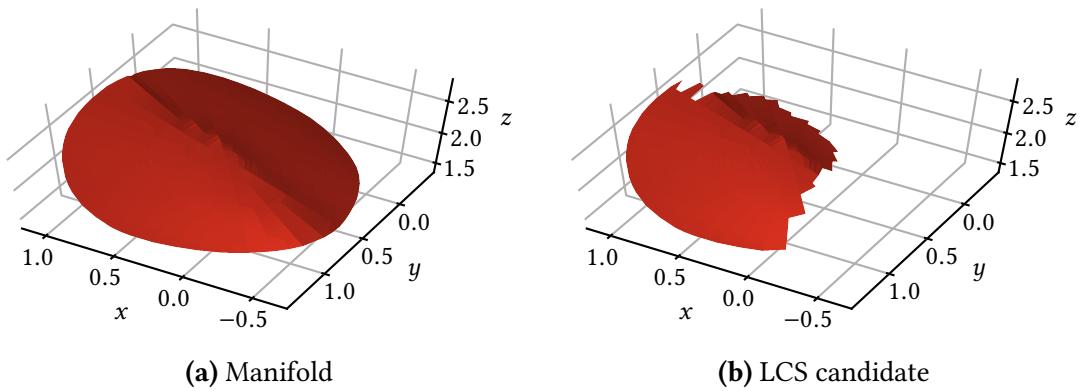


**Figure 4.3:** Numerical convergence as  $\text{min\_sep}$  decreases



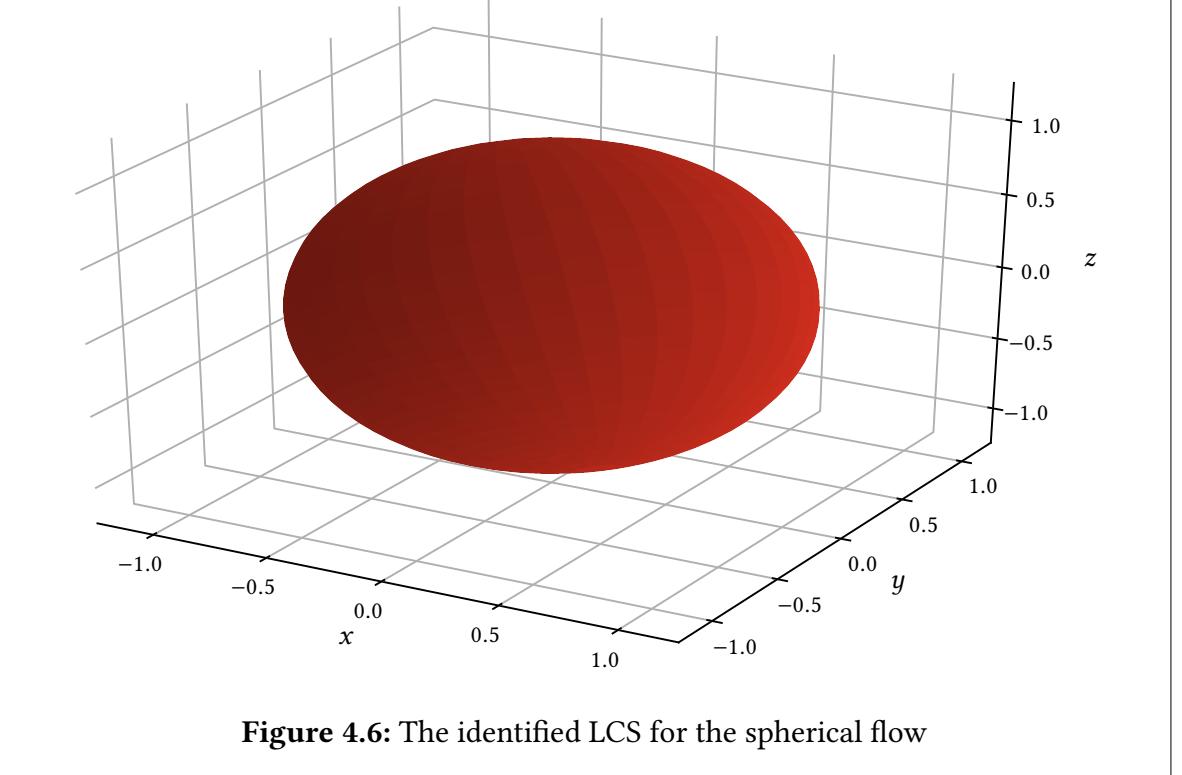
(a) Trajectories forced radially outwards      (b) Pure linear combinations of  $\xi_1$  and  $\xi_2$

**Figure 4.4:** Trajectories with local directionality given by linear combinations of  $\xi_1$  and  $\xi_2$ .

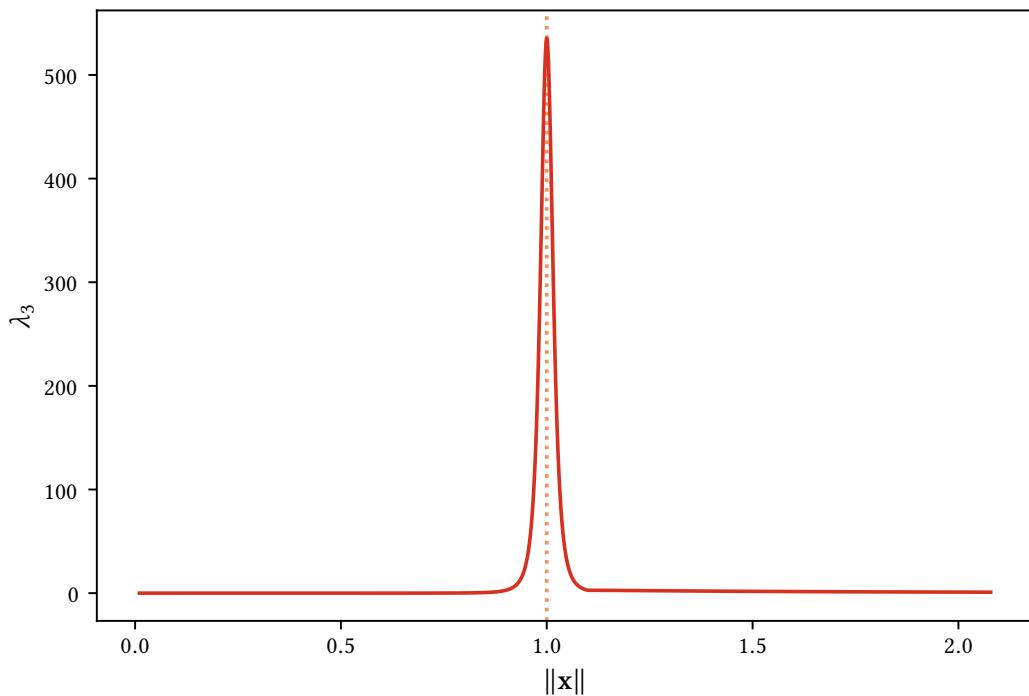


(a) Manifold      (b) LCS candidate

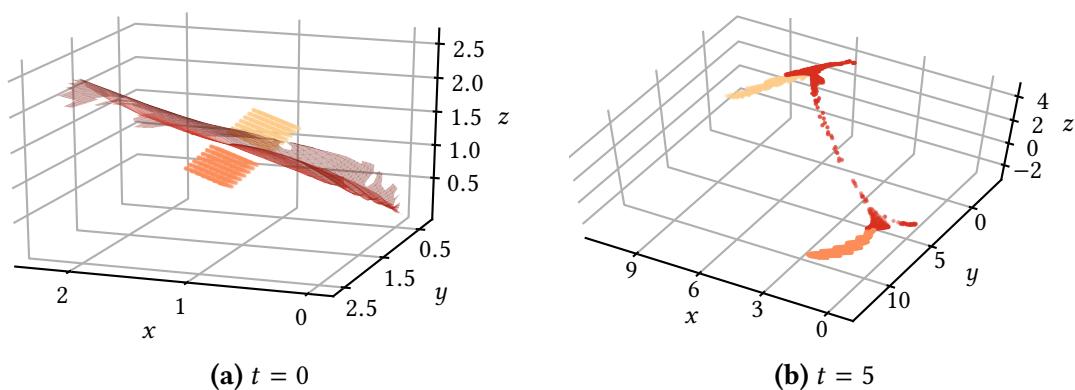
**Figure 4.5:** Converting a manifold to an LCS candidate



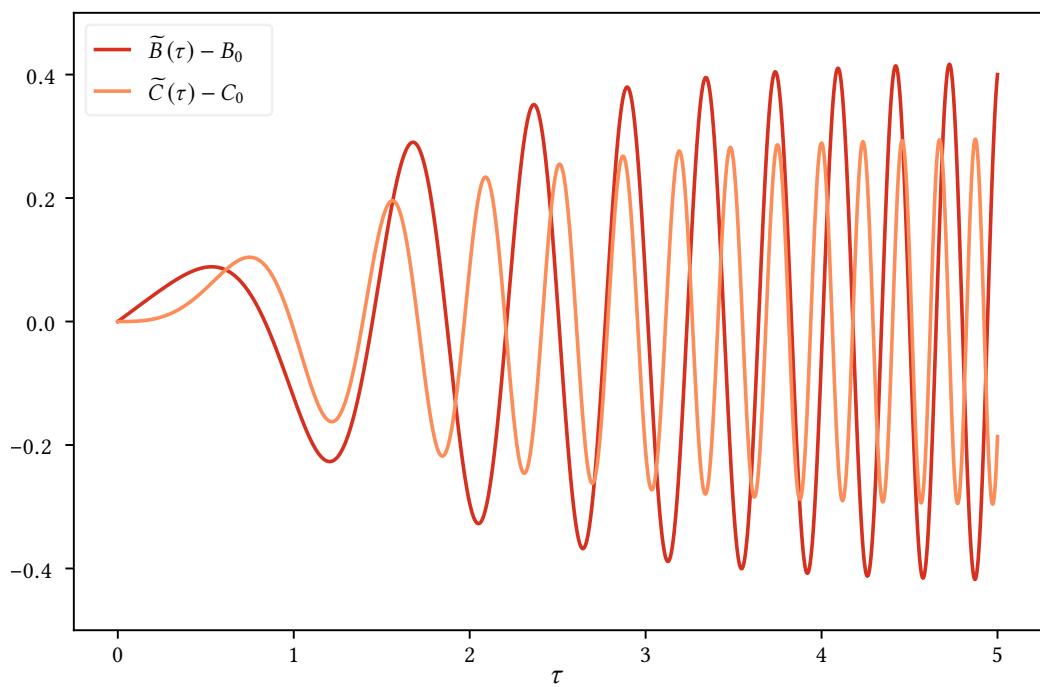
**Figure 4.6:** The identified LCS for the spherical flow



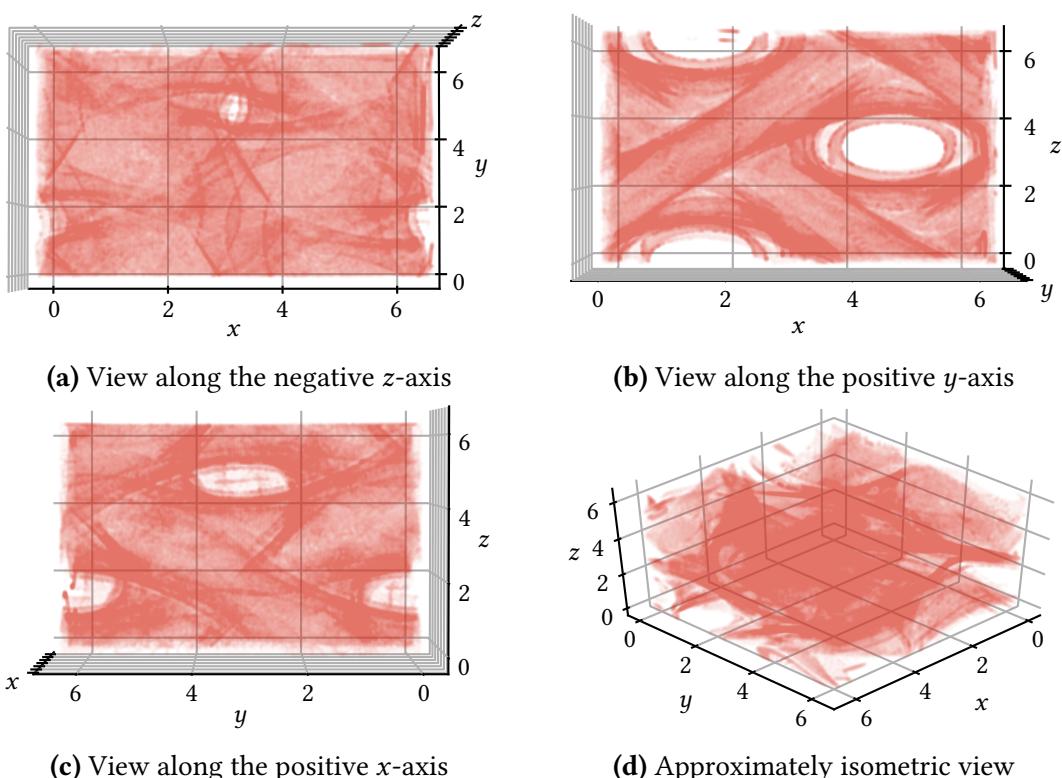
**Figure 4.7:**  $\lambda_3$  as a function of radius, for the spherical flow. Displayed here is the arithmetic average across all solid angles.



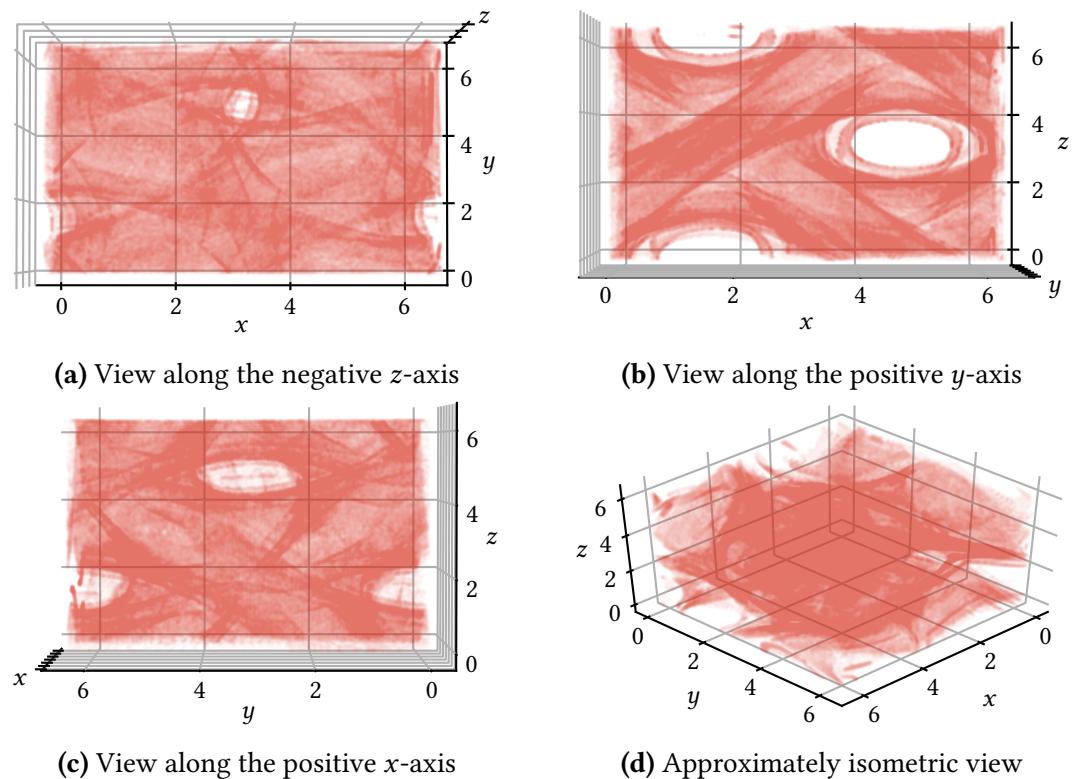
**Figure 4.8:** Blob test for verifying expected LCS behaviour



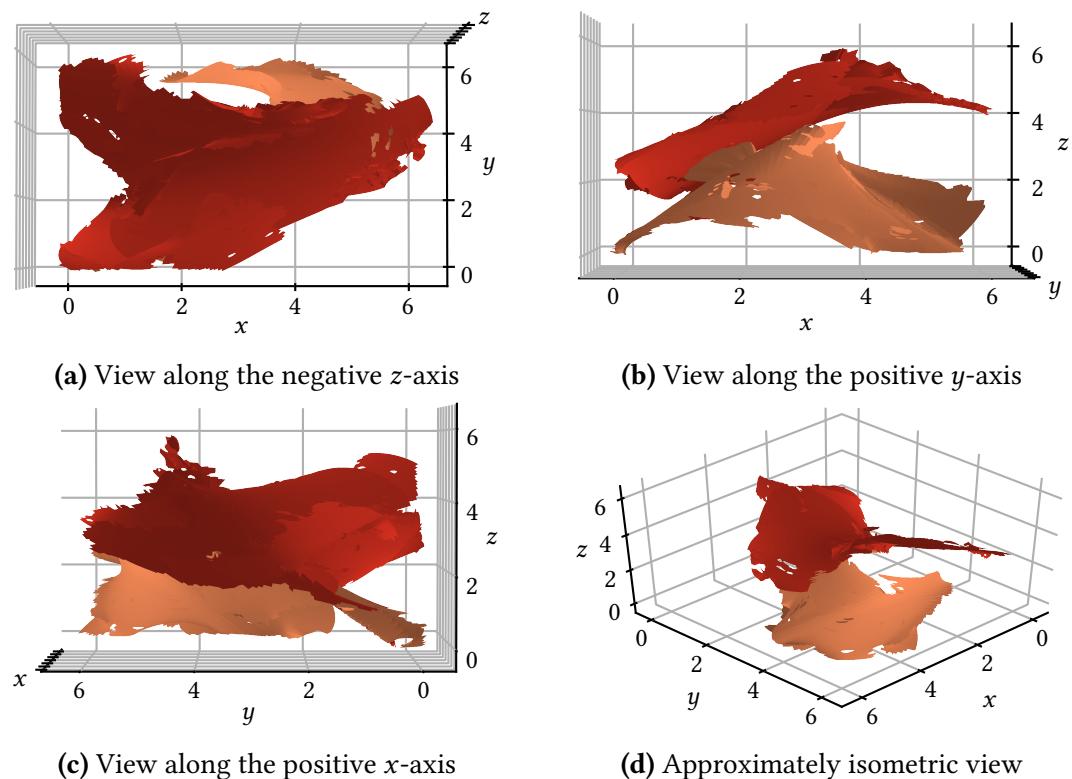
**Figure 4.9:** Time evolution of nonstationary ABC coefficients



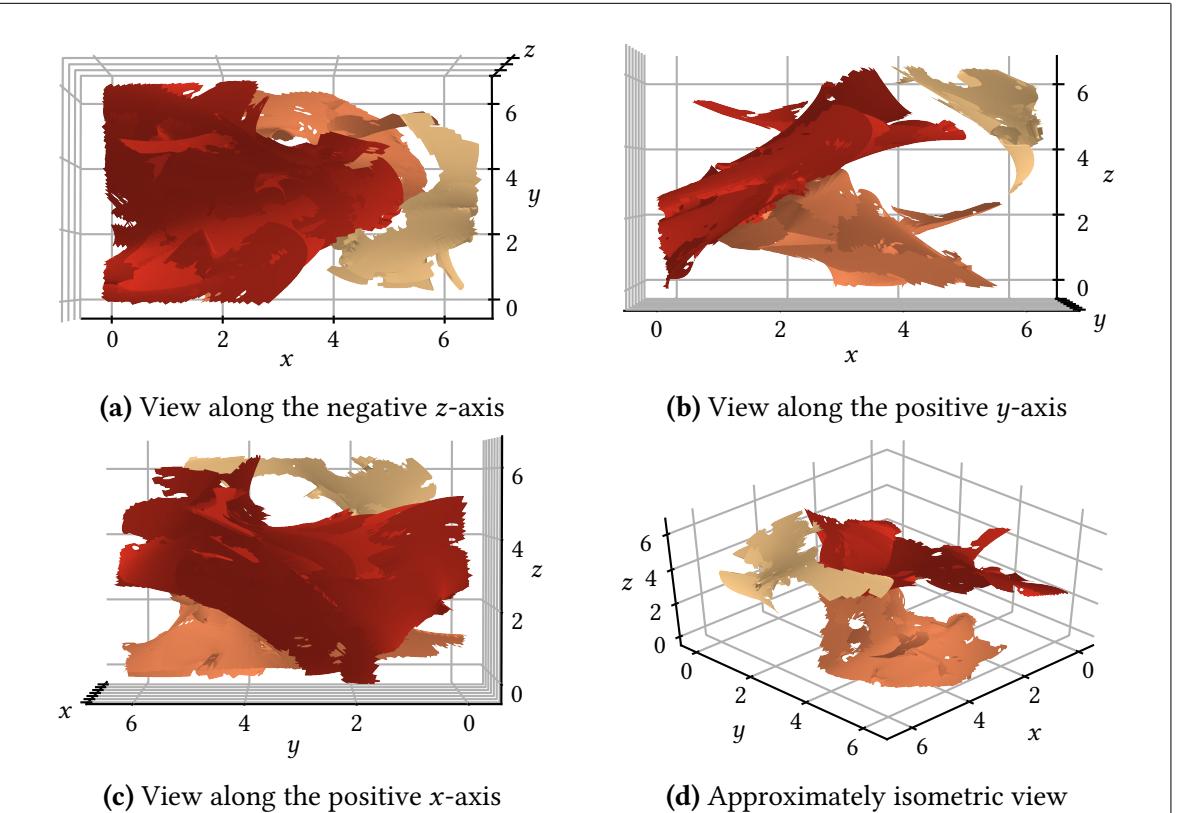
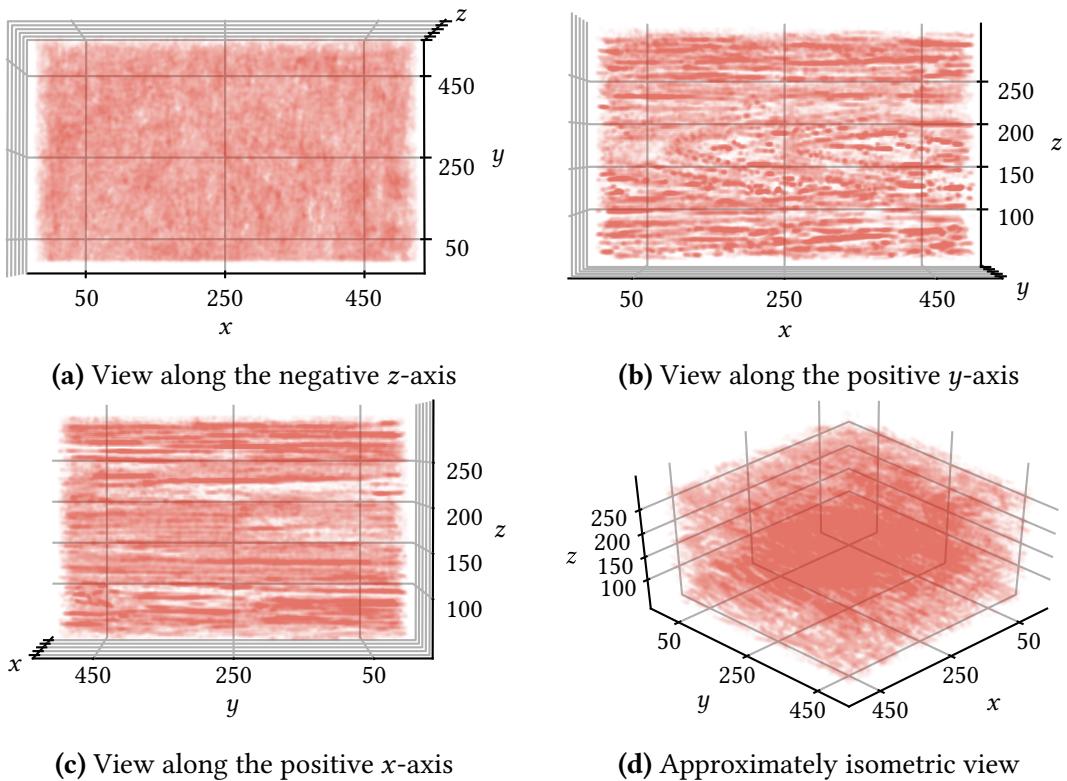
**Figure 4.10:** ABD domain obtained for stationary flow.  $\epsilon = 0.005$ .

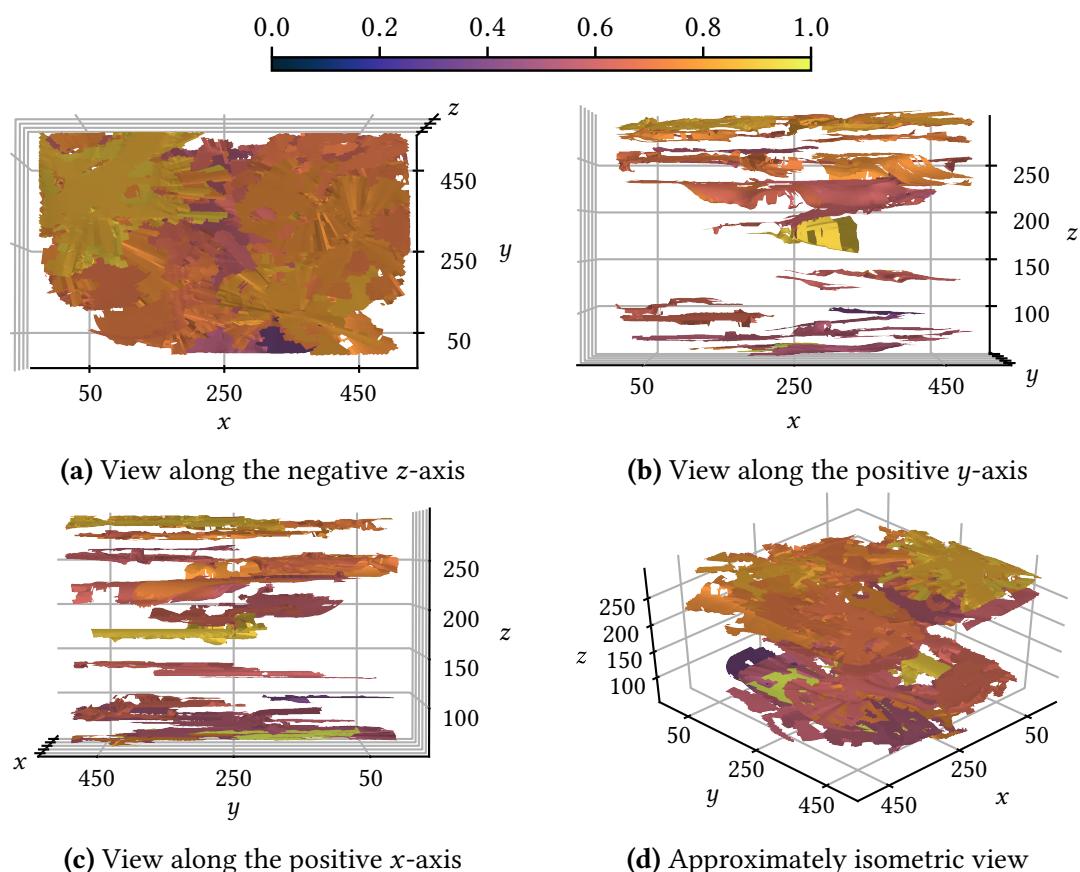


**Figure 4.11:** ABD domain obtained for dynamic flow.  $\epsilon = 0.005$ .



**Figure 4.12:** LCSs obtained for stationary ABC flow

**Figure 4.13:** LCSs obtained for dynamic ABC flow**Figure 4.14:** ABD domain obtained for gridded data. The axes have dimension meters.  $z = 0$  corresponds to the sea level, increasing downwards.  $\epsilon = 1$ .



**Figure 4.15:** LCSs obtained for stationary ABC flow

---

---

## **5 Discussion**

---



---

---

## **6 Conclusions**

---



## References

- Farazmand, M. and Haller, G. (2012a). "Computing Lagrangian coherent structures from their variational theory". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 22.1, p. 013128. ISSN: 1054-1500. DOI: [10.1063/1.3690153](https://doi.org/10.1063/1.3690153).
- Farazmand, M. and Haller, G. (2012b). "Erratum and addendum to 'A variational theory of hyperbolic Lagrangian coherent structures' [Physica D 240 (2011) 547–598]". In: *Physica D: Nonlinear Phenomena* 241.4, pp. 439–441. ISSN: 0167-2789. DOI: [10.1016/j.physd.2011.09.013](https://doi.org/10.1016/j.physd.2011.09.013).
- Frisch, U. (1995). *Turbulence, the legacy of A.N. Kolmogorov*. 1st ed. Cambridge University Press. ISBN: 0-521-45103-5.
- Hairer, E., Nørsett, S. P., and Wanner, G. (1993). *Solving Ordinary Differential Equations I: Nonstiff Problems*. 2nd ed. Springer-Verlag Berlin Heidelberg. ISBN: 978-3-540-56670-0. Corrected 3rd printing, 2008.
- Hairer, E. and Wanner, G. (1996). *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. 2nd ed. Springer-Verlag Berlin Heidelberg. ISBN: 978-3-642-05221-7. Corrected 2nd printing, 2002.
- Haller, G. (2010). "A variational theory of hyperbolic Lagrangian Coherent Structures". In: *Physica D: Nonlinear Phenomena* 240.7, pp. 547–598. ISSN: 0167-2789. DOI: [10.1016/j.physd.2010.11.010](https://doi.org/10.1016/j.physd.2010.11.010).
- Karrasch, D. (2012). "Comment on 'A variational theory of hyperbolic Lagrangian coherent structures, Physica D 240 (2011) 574–598'". In: *Physica D: Nonlinear Phenomena* 241.17, pp. 1470–1473. ISSN: 0167-2789. DOI: [10.1016/j.physd.2012.05.008](https://doi.org/10.1016/j.physd.2012.05.008).
- Oettinger, D. and Haller, G. (2016). "An autonomous dynamical system captures all LCSs in three-dimensional unsteady flows". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 26.10. ISSN: 1054-1500. DOI: [10.1063/1.4965026](https://doi.org/10.1063/1.4965026).
- Onu, K., Huhn, F., and Haller, G. (2015). "LCS Tool: A computational platform for Lagrangian coherent structures". In: *Journal of Computational Science* 7, pp. 26–36. ISSN: 1877-7503. DOI: [10.1016/j.jocs.2014.12.002](https://doi.org/10.1016/j.jocs.2014.12.002).
- Zhao, X.-H. et al. (1993). "Chaotic and Resonant Streamlines in the ABC flow". In: *SIAM Journal on Applied Mathematics* 53.1, pp. 71–77. DOI: [10.1137/0153005](https://doi.org/10.1137/0153005).



---

---

## A Appendix A