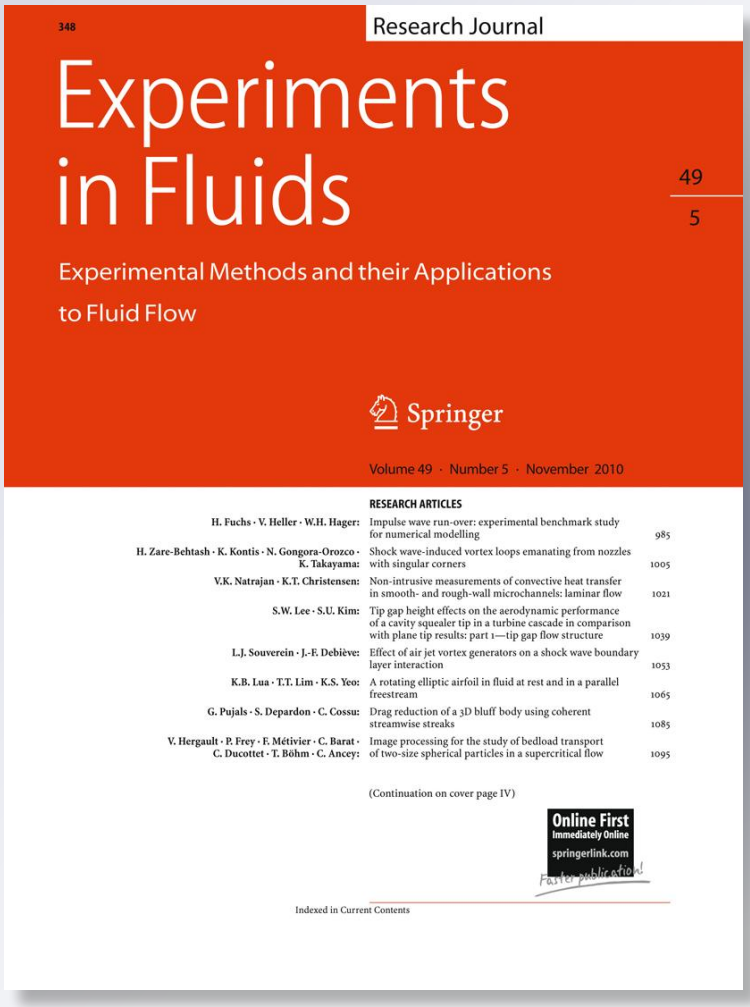
Discretization of the vorticity field of a planar jet

Experiments in Fluids
Experimental Methods and their Applications to Fluid Flow

ISSN 0723-4864
Volume 49
Number 5

Exp Fluids (2010)
49:1161-1175
DOI 10.1007/
s00348-010-0862-8



Springer

Exp Fluids (2010) 49:1161–1175
DOI 10.1007/s00348-010-0862-8

RESEARCH ARTICLE

Discretization of the vorticity field of a planar jet

Natalie Ross · Jean Hertzberg · Elizabeth Bradley

Received: 6 February 2009 / Revised: 1 November 2009 / Accepted: 4 March 2010 / Published online: 1 April 2010

Abstract In data assimilation, information from sensors is used to correct the state variables of a numerical model. This has been used to great advantage by the weather prediction community in the context of direct numerical simulation (DNS) models, but has seen comparatively little use in point-vortex models. This is due in large part to data-processing issues. In order to keep up with the speeds necessary for effective data assimilation, one must extract and discretize the vortex structures from velocity field data in a computationally efficient fashion—i.e., using as few discrete vortices as possible to model the measured flow. This paper describes a new strategy for accomplishing this and evaluates the results using data from a laboratory-scale vortex-dominated planar jet. Large-scale vortex structures are found using a family of variants on traditional vortex extraction methods. By augmenting these methods with simple computational topology techniques, one obtains a new method that finds the boundaries of the coherent structures in a manner that naturally follows the geometry of the flow. This strategy was evaluated in the context of two standard vortex extraction methods, vorticity thresholding and Okubo–Weiss, and tested upon velocity field data from the experimental fluid flow. The large-scale structures found in this manner were then modeled with collections of discrete vortices, and the effects of the grain size of the discretization and the parameters of the discrete vortex model were studied. The results were evaluated by comparing the instantaneous velocity field induced by the discrete vortices to that measured in the jet. These comparisons showed that the two extraction techniques were comparable in terms of sensitivity and error, suggesting that the computationally simpler vorticity thresholding method is more appropriate for applications where speed is an issue, like data assimilation. Comparisons of different discretization strategies showed that modeling each large-scale vortex structure with a single discrete vortex provided the best compromise between mean-squared error and computational effort. These results are of potential interest in any situation where one must balance accuracy and expense while extracting vortices from a snapshot of a flow field; data assimilation is only one example.

N. Ross · E. Bradley (✉)
Department of Computer Science, University of Colorado,
Boulder, CO 80309-0430, USA
e-mail: lizb@cs.colorado.edu

J. Hertzberg
Department of Mechanical Engineering, University of Colorado,
Boulder, CO 80309-0427, USA

1 Introduction

The motivation behind the work described in this paper is the use of sensor data to correct the state variables of a point-vortex model of a planar jet flow. This kind of procedure, known as data assimilation, has been used extensively for many years by the numerical weather prediction community, but only recently in conjunction with point-vortex models, and then only in simulations involving perfect information about ideal point vortices (Ide and Ghil 1997; Ide et al. 2002). Point-vortex (PV) models have different strengths and weaknesses than the kinds of traditional DNS models that have been used in data assimilation research. Because they do not use fixed grids, their state variables can be tailored to the flow, effectively lowering their computational complexity.[1] However,

[1] Data assimilation has been used in adaptive grid models, but only in idealized cases, and the associated strategies are complex and computationally intensive (Anderson 2010).

significant processing effort can be required to extract values from the data for use in setting up and correcting their state variables.

This paper is an exploration of these issues and trade-offs. It addresses the specific issues involved in preprocessing an *instantaneous* snapshot of a velocity field for use in correcting a running PV model *at a single point in time*. Because both accuracy *and speed* are important issues in correcting a running solver, and because corrections are delivered repeatedly, the type of fine-grained discretization that is typically required to set up the proper conditions for temporal convergence of the model, such as Cottet and Koumoutsakos (2000, pp. 22–31) or Hald (1991), is not a good solution here. Rather, this problem demands an intelligent, easily adaptable discretization: one that models snapshots of the flow with only as many discrete vortices as are demanded by the situation at hand, and one that is guided by the geometry of the flow and not solely by an arbitrary grid.

The solutions described in this paper are based on two classic vortex extraction methods: vorticity thresholding and the Okubo–Weiss criterion. A computational topology technique (Robins et al. 1998, 2004a, b) was incorporated into these methods, allowing them to find the vortex boundaries in a principled manner that follows the smoothness properties of the flow. These techniques were evaluated in the context of particle image velocimetry (PIV) data from a planar air jet, using three different strategies for distributing discrete vortices to match the characteristics of each large-scale vortex. The results appeared to be effective for the purposes of data assimilation into a point-vortex model, as described in Ross (2008), but much more work remains to be done to compare the overall results to traditional DNS-based assimilation schemes. This comparison is not trivial: the underlying solvers are fundamentally different, as are the data required to correct them and the correction strategies used to do so. (DNS models respond very differently to step changes in state variables, for instance, than do PV models.) This broader comparison is outside the scope of this paper; the focus here is on how to process data for the purposes of assimilation into PV models, not on assimilation strategies or the dynamics of models that use those strategies.

Extracting vortex positions and strengths from velocity field data are important for a variety of reasons. Vortices are surprisingly hard to define, however, and experimental data are inevitably noisy, so this is not a trivial problem. Some of the methods that have been developed to work around these challenges identify vortices and distinguish them from other types of coherent structures; others simply discretize the instantaneous vorticity field. The work of Jeong and Hussain (1995) provides a cornerstone for much of the debate in the fluids literature about this topic, along with a useful definition that decomposes the velocity gradient tensor $\nabla\vec{u}$ into symmetric (S) and anti-symmetric (Ω) parts:

$$\Omega = \frac{1}{2}\left[(\nabla\vec{u}) - (\nabla\vec{u})^T\right]$$

$$S = \frac{1}{2}\left[(\nabla\vec{u}) + (\nabla\vec{u})^T\right]$$

Ω is an effective measurement of vorticity in an incompressible flow. Thus, searching for regions in which the norm of Ω dominates the norm of S can be an effective technique for identifying vortices; this is the Q-criterion of Hunt et al. (1988). The Okubo–Weiss method (Okubo 1970; Weiss 1991) is a two-dimensional version of the Q-criterion that identifies regions where the squared rate of rotation, $|\Omega|^2$, dominates the squared rate of strain, $|S|^2$. In these regions, the flow behavior is classified as elliptic in nature; outside them, the fluid motion is hyperbolic. It is worth mentioning that Haller's method (2005) also identifies elliptical regions and is invariant under *any* type of coordinate change. Haller's method sprinkles passive tracers throughout the flow, defining vortices as regions where all tracer trajectories remain on elliptic paths. This careful distinction between elliptic and hyperbolic properties is important if one wants to distinguish vortices from other vorticity-bearing structures such as shear layers, but that level of effort is not warranted in the application treated here, where the goal is simply to discretize a snapshot of the vorticity field. A variety of other approaches use $\nabla\vec{u}$ in different ways; (Adrian et al. 2000; Chong et al. 1990), for instance, take its imaginary eigenvalues as evidence of local swirling motion. Some vortex extraction research attempts to distinguish vortices from other structures that have high vorticity, such as shear layers (e.g., Pemberton et al. 2002; Vollmers 2001). Other methods use the geometry of the flow velocity directly, searching for foci and centers in streamline patterns (Lugt 1979); minima in local-angle gradients (Pemberton et al. 2002); spanwise-aligned regions that are highly correlated (Scarano et al. 1999); connecting regions where the velocities are in opposite directions (Vollmers 2001); and regions where the finite-time Lyapunov exponents of the flow suggest the presence of so-called Lagrangian coherent structures (Shadden et al. 2005); among others. Many groups have worked out ways to fit velocity data to various analytical forms, such as wavelets (Camussi 2002; Farge et al. 1999; Seigel and Weiss 1997) or orthogonal and Fourier decompositions, such as Farge et al. (2003), Palacios et al. (1996) and Preisendorfer (1988), and then use those decompositions to find the vortices. Other groups use predictor-corrector methods (Banks and Singer 1995) or even neural nets (Joseph et al. 1964) to find coherent structures in velocity data.

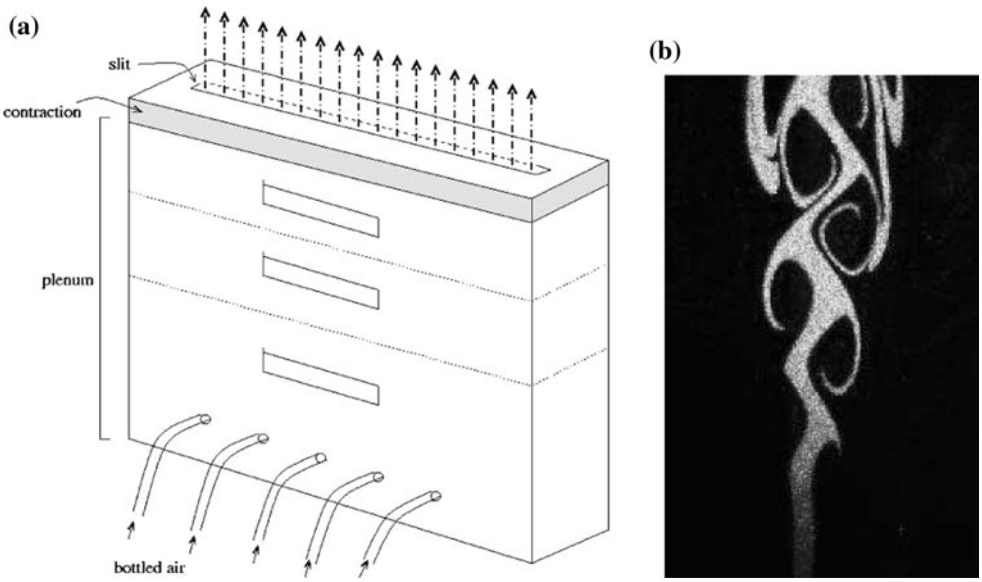


Fig. 1 A planar air jet: **a** sketch of the experimental apparatus. **b** Side view of the jet flow with 16.83 Hz forcing. $Re \approx 93$

The evaluation case used in this paper is a planar flow, and the goal of the methods described in the following sections is to compute the positions and strengths of a set of discrete vortices whose induced velocity field matches the measured flow at a particular instant in time. In contrast to some of the methods mentioned above, there is no need to distinguish between vortices and shear layers or other vorticity-bearing structures. Deriving initial conditions that are good enough to assure long-term temporal convergence is unnecessary because correction is ongoing and periodic; the goal is simply to discretize the instantaneous vorticity field, and in a manner that can easily be adapted to the precision requirements of different modeling situations. The controlled laboratory environment is particularly important here, given the complex, non linear nature of the flow, the model, the extraction methods, and (ultimately) the assimilation dynamics. A planar jet in the lab is of course far simpler—and far easier to measure—than the complex 3D flows that dominate numerical weather prediction problems. Nonetheless, a careful, principled study of how extraction and assimilation work in this laboratory flow can potentially generalize beyond this particular system, usefully informing how—and whether—data assimilation works in point-vortex models. And these results are of potential interest in any other situation where one must balance accuracy and expense while extracting vortices from a snapshot of a flow field.

2 Methods

2.1 Apparatus

A sketch of the experiment is presented in Fig. 1a. Filtered air entered the base of a plenum, whose internal cross-section was 400 mm by 15 mm, and was passed through three flow treatment screens of 1-mm mesh spacing with 0.2-mm-diameter wire. A cubic spline contraction of area ratio 6–1 formed the top of the plenum. Flow emerged from an exit slit 400 mm long and 2.5 ± 0.01 mm wide, giving an aspect ratio of 160:1. The velocity profile at the jet exit was parabolic and developed into a plane symmetric Bickley jet within a few nozzle widths downstream. The peak jet velocity was 1.36 m/s, corresponding to a Reynolds number of 93 based on the nozzle half-width. The downstream direction is referred to as x, transverse as y, and spanwise as z. The system was highly sensitive to mechanical vibrations and motion of the ambient air. To minimize these effects, the entire experiment stood on vibration control mounts and was semi-enclosed by a 1 m^3 plexiglass box. The resulting jet had a turbulence intensity, in the absence of any forcing, of less than 0.5%. The apparatus and its flow properties are described in more detail in Peacock et al. (2004). For the study presented here, a single external loudspeaker was used to excite the antisymmetric mode.[2] This was accomplished by driving the loudspeaker at the natural frequency of the jet (16.83 Hz). A laser-sheet visualization of the jet in its antisymmetric mode is displayed in Fig. 1b. (Seeding is described below.) Note that vortices appear well-defined in this flow, making it a good candidate for point-vortex modeling.

Velocity data were acquired using particle image velocimetry (PIV; Raffel et al. 1998). The jet was seeded with theater fog, consisting of a water/glycerin mixture condensed into ≈1-micron-diameter droplets. Illumination

[2] In Peacock et al. (2004), the symmetric and antisymmetric instabilities of the jet were excited by MEMS flaps mounted on either side of the nozzle.

was provided by a New Wave Solo PIV Nd:YAG laser at medium power. Images were captured using a TSI camera, model PIVCAM 13-8, with 1,280 × 1,024 12-bit pixels. PIV processing was carried out with TSI Insight software. The field of view spanned 15 to 22 jet widths in the streamwise (x) direction. Data were acquired at 16 specific phases of the loudspeaker excitation, and 420 realizations were averaged for each phase, ensuring statistical convergence (Farrell 2008). The resulting velocity field of 61 × 65 vectors had a uniform grid spacing of $h = 0.87$ mm. All lengths reported in this paper are scaled by this grid spacing; all velocities are scaled by the maximum velocity in the corresponding flow-field snapshot.

2.2 PIV results

The left-hand column of Fig. 2 presents velocity field data for the jet at four different phases in the flow cycle, ~15 milliseconds apart. For clarity, all vector plots in this paper are downsampled: only 1/4 of the vectors are shown. The right-hand column of Fig. 2 shows the corresponding vorticity fields, computed from the velocity fields using the center difference method, with forward and backward differences at the edges of the domain, as needed. The maximum velocities in Fig. 2a and d, which are analyzed in more depth in the following section, were 1.22 and 1.16 m/s, respectively. (These values are slightly lower than the maximum nozzle velocity because those snapshots were taken a few centimeters downstream from the nozzle.) The goal of the methods described in this paper is to compute a distribution of discrete vortices that model these instantaneous snapshots of the field.

2.3 Analysis

The methods presented here are modifications of two well-known vortex extraction techniques: vorticity thresholding and the Okubo–Weiss method. Both were designed to identify regions of high vorticity in data like the snapshots in the right-hand column of Fig. 2. A computational topology technique was incorporated into both methods in order to allow them to find vortex boundaries in a manner that matches the smoothness properties of the flow. In particular, each high-vorticity region was assumed to be a *connected component* and was found by "growing" the classification outwards from a single starting point. The method that combines vorticity thresholding and this notion of connectedness is covered in Sect. 2.3.1; Sect. 2.3.2 describes how to incorporate connectedness into Okubo–Weiss. Once these large-scale high-vorticity regions were identified, the next task was to model them with a collection of discrete vortices whose induced velocity fields model the original PIV data snapshot. The discretization strategies used to accomplish this are described in Sect. 2.3.3. The evaluation process is described in Sect. 2.4.

2.3.1 *Connected vorticity thresholding*

The connected-component/vorticity thresholding method (hereafter "connected vorticity thresholding") uses the following strategy to find large-scale vortices in gridded vorticity data:

1. Take the magnitude of the vorticity field and find its maximum value. Label this grid point (x_0, y_0).
2. Let $\mathbf{M}$ represent the set of points in a particular vortex. Initially, set $\mathbf{M} = \{(x_0, y_0)\}$. Then, starting from the point at (x_0, y_0), construct a connected component, consisting of neighboring points whose vorticity magnitude is above a threshold T. For the purposes of this method, a point is considered connected to four "neighbors"—one above, one below, one to the left, and one to the right: $(x_0, y_0 + 1)$, $(x_0, y_0 - 1)$, $(x_0 - 1, y_0)$, and $(x_0 + 1, y_0)$. Check each of these points and add it to $\mathbf{M}$ if its vorticity exceeds T. (Issues regarding the choice of T are discussed below.)
3. Repeat Step (2), recursively checking the neighbors of those neighboring points and including them in the component if their vorticity magnitudes exceed the threshold. Terminate when no new neighbor points meet the criterion. (This is the edge of the component.)
4. Compute the average vorticity over all points $\mathbf{M}$ in the connected component. Multiply this by the area covered by those points to get an approximate circulation, or strength, for the associated large-scale vortex structure.
5. Remove the points identified in Steps (2) and (3) from consideration and repeat the process starting from Step (1) to find the next large-scale vortex.

The threshold parameter T in Step (2), some version of which is used in virtually every other instantiation of the Okubo–Weiss method, controls how much vorticity is lumped into each vortex. It is intended to be used by the modeler to tune the extraction process to the requirements of his or her problem. The general challenge in isolating coherent structures with any vorticity thresholding method is to choose a T value that causes the connected component algorithm to include as much area as possible for each structure without accidentally grouping two distinct ones together. This is discussed later in this paper. The main series of results presented in Sect. 3 uses $T = 0.14|\max(\omega) - \min(\omega)|$, where $\max(\omega)$ and $\min(\omega)$ are the maximum and minimum vorticity values over all grid points in the data set.

Thresholds are always arbitrary, of course, as are definitions of what constitutes a "distinct" vortex, and

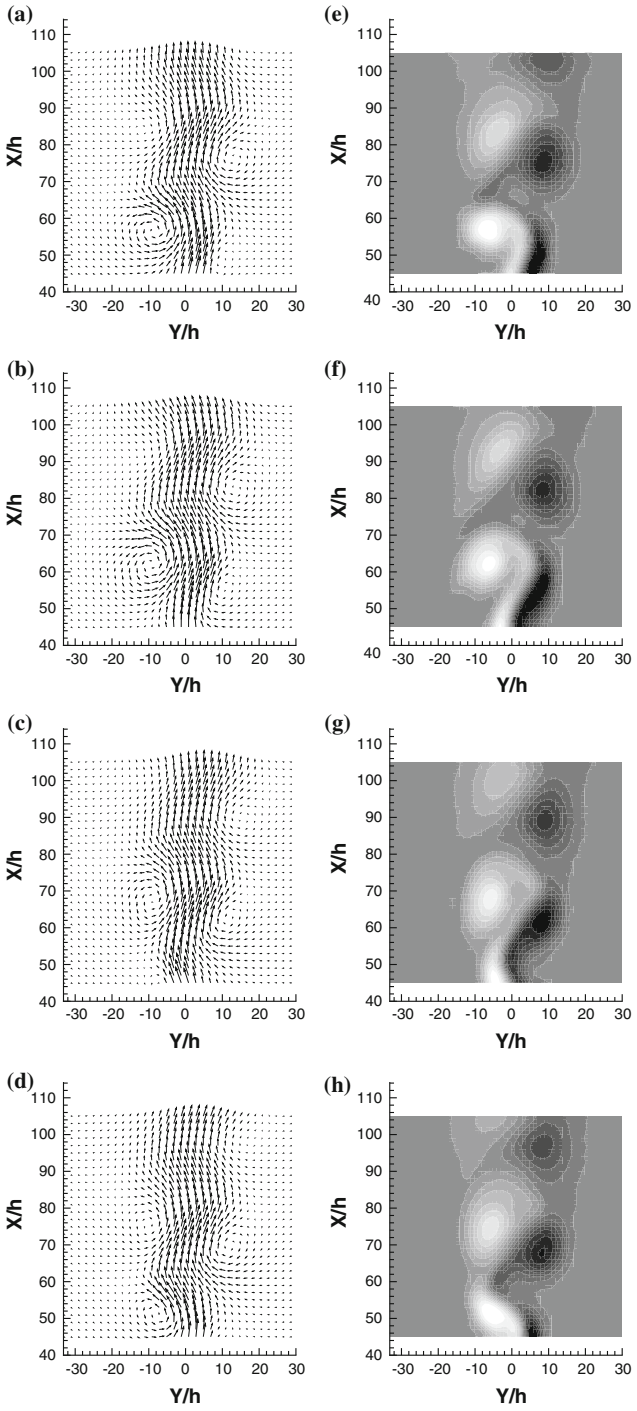


Fig. 2 Velocity and vorticity fields of the planar jet under periodic forcing. Images **a**–**d** show PIV data at various phases of the drive period: **a** 22.5°, **b** 90°, **c** 202.5° and **d** 292.5°. Images **e**–**h** show the corresponding vorticity fields. All velocity field plots in this paper are downsampled for clarity

computer interpretation of images often falls far short of what expert eyes can do. Tuning T allows a user of this method to coarsen the discretization at will and at need, in a manner that combines the features of simple thresholding and the notion of a connected component. Thresholding effects play out in slightly different ways here than in traditional vorticity thresholding methods, though, because of that connected-component step; this is also discussed later in this paper.

2.3.2 *Connected Okubo–Weiss*

The Okubo–Weiss criterion (Okubo 1970; Weiss (1991) is more complex and more computationally intensive than vorticity thresholding—and arguably more reliable, in the sense that it is invariant in translating frames of reference (Galilean invariance). The original Okubo–Weiss criterion defines a vortex as any region where $W = |\Omega|^2 - |S|^2 > 0$. The variant presented here, termed "connected Okubo–Weiss," modifies that approach in two ways. First, it allows the user to specify a threshold for the inequality—e.g., to consider only points where $|\Omega|^2$ is *significantly* greater than $|S|^2$. Second, it again uses the notion of a connected component to define the boundaries of the vortices:

1. Compute $W = |\Omega|^2 - |S|^2$. Remove from consideration any grid points (a, b) for which
   $$W_{(a,b)} \leq k[\max(W)]$$
2. Find the grid point with the maximum positive value of W.
3. Using the same recursive neighbor-checking algorithm described in Step (3) of the algorithm in Sect. 2.3.1, compute the connected component starting from the point identified in Step (2).
4. Compute the average vorticity for the points identified in Step (3). Multiply this by the area covered by those points to get an approximate circulation, or strength, for the associated vortex.
5. Remove the points identified in Step (3) from consideration and repeat the process starting with Step (2) to find the next vortex.

Like the method of Sect. 2.3.1, this method has a tuning parameter: the cutoff k. This parameter is used not only to filter out the points where rotation dominates strain, as in classic Okubo–Weiss, but also to ignore points where strain and rotation rates are similar—i.e., points that fall near the sharp threshold that is embodied in the Okubo–Weiss criterion. Setting $k = 0$ is equivalent to the original Okubo–Weiss mathematics, where the smallest difference between rotation and strain defines a vortex. If instead one wants to enforce a larger separation, one can use k to exclude points where that difference falls below some fraction of the maximum W value. This, like the T parameter in Sect. 2.3.1, is a common modification of this method; its use here is not novel and the associated issues are well known. Again, no arbitrary fixed threshold is effective for all modeling problems. The main series of results in Sect. 3

uses $k = 0.023$. This means that values in the lowest 2.3% of the maximum W in the field, where $|\Omega|^2$ is only slightly larger than $|S|^2$, are not considered to be conclusive evidence—one way or the other—of a vortex. Because the mathematics behind this tuning parameter are different than the simpler notion of a vorticity threshold, and because of the connected-component step in the method, the reasoning involved in its tuning is also somewhat different, as described later in this paper.

The Okubo–Weiss method has been shown to be inaccurate if the velocity gradient tensor is time-varying (Segur 1998), and there are higher-order corrections to the method for this; see Hua and Klein (1998) and or (Hua et al. 1998). The basic version of the criterion is used here because the additional computational cost involved in these corrections is not justified for the problem treated in this paper. In other applications, where tracking variations in time and space dominate the modeling requirements, these accuracy issues may be important, warranting the inclusion of higher-order corrections or even different extraction techniques.

2.3.3 Vortex discretization

Any decomposition of a vorticity field snapshot into discrete vortices is an approximation—one whose accuracy depends on the number of discrete vortices involved. And, as in any modeling problem, it involves a trade-off: one wants to use *only* as many discrete vortices as are necessary for the requirements of the problem, distributing them such that their superposition models the larger-scale vortex structures that are present in the field, to within some specified accuracy. The methods described here are designed with data assimilation in mind: they preprocess static snapshots of the velocity field for use in correcting the state variables of a point-vortex solver. As such, the requirements upon them are different than in applications that demand a discretization that is accurate enough to assure temporal convergence of the model when started from those initial conditions (Barba 2007; Barba et al. 2005). Indeed, that kind of fine-grained discretization could pose problems for the assimilation process, even with modern acceleration techniques (Cottet and Koumoutsakos 2000) (Appendix B), since each discrete vortex adds at least three state variables.[3] to the model.

In order to assess these effects and trade-offs, three discretization strategies were evaluated:

(i) a fine-grained method that places discrete vortices of appropriate strength at every grid point, as is common in simulations of turbulence and mixing,

(ii) a coarse-grained distribution that places a single discrete vortex of appropriate strength at the local maximum of vorticity of each large-scale vortex, and

(iii) a parametrized method that distributes discrete vortices of *uniform* strength across each large-scale vortex under the control of a discretization parameter, as described in the following paragraphs.

Strategies (ii) and (iii) work with the connected components found by the methods described in Sects. 2.3.1 and 2.3.2; strategy (i) works with the raw vorticity field, placing a discrete vortex at every grid point and setting its strength equal to the circulation in the corresponding grid cell. Assigning a single discrete vortex per connected component is straightforward; strategy (ii) does this using the locations and strengths found in Steps (1) and (4) of the method that computed the component. The parametrized discretization strategy (iii) distributes uniform-strength discrete vortices across each connected component, as described in the steps below. The strength quantum γ_0 used in this procedure is determined by the strength $\Gamma_{\min}$ of the weakest vortex in the flow, via a quantization parameter q: $\gamma_o = \Gamma_{\min}/q$. The steps of the method are as follows:

1. Find the maximum magnitude of the circulation in the connected component.
2. If there is not already a discrete vortex at the location of the maximum found in Step (1)—i.e., if this is the first time that the algorithm has examined this point—place one there, choosing the sign to match the sign of the circulation. If there is, find the closest uncovered point (by Euclidean distance) and place the discrete vortex there.
3. Subtract the quantum of this discrete vortex from the circulation at the location in Step (2) and repeat from Step (1).

An alternate version of the method that uses the centroid instead of the vorticity maximum was also tested, but the one in steps 1–3 above proved to be more accurate. The parameter q controls the grain of the discretization in the obvious way. The main series of results in Sect. 3 uses $q = 2$, for instance, which assigns two discrete vortices to the weakest large-scale vortex. Dividing the circulation among multiple discrete vortices has potential advantages outside the context of matching a single snapshot of the field, as it can allow the time-stepped point-vortex equations to model vortex core deformation more effectively. The use, here, of *uniform* strength vortices is in accordance with early random-walk vortex techniques (Gustafson and Sethian 1991), but is in contrast to modern blob methods (Cottet and Koumoutsakos 2000). Recall that the goal here is not to produce

[3] Position and strength.

initial conditions that allow the point-vortex model to track the further temporal evolution of the flow as accurately as is the goal in those modern methods. The situation here is different: the future evolution of the modeled flow is a hybrid process, controlled by a combination of the model equations and periodic correction performed by the data assimilation system. As such, there is no need for careful attention in every extraction step to future temporal convergence. Indeed, that would be an unwarranted cost, since the periodic assimilation is designed to correct any disparities between the model state and the flow field at later points in time.

2.4 Evaluation

Given a static flow field, the methods in the previous two sections compute the locations and strengths of a collection of discrete vortices. To evaluate the results, the velocity field induced by each vortex configuration was calculated and then compared to the original PIV velocity field. The velocity fields were reconstructed from the vortex positions and strengths in the usual way, using Biot-Savart. For the results presented in Sect. 3, the velocity at each grid point was calculated by summing the velocities induced there by each discrete vortex, computed using the blob model equations (Chorin 1973) with simple solid-body rotation in the core. This model was chosen to avoid the large non-physical velocities that the singularity in the point-vortex model would produce in the fields of Fig. 2, where the approximate diameter of the viscous cores of the vortices is 10 times the grid spacing. The square of pointwise difference between the induced field and the original PIV field was then calculated and averaged across all grid points. The optimal core radius r_c—i.e., the r_c that minimized the mean-squared error between measured and induced velocity fields—was found individually for each vortex configuration. The overall circulation of the induced field was also calculated.

3 Results and discussion

3.1 Extraction

Figure 3 shows the results of applying the connected vorticity thresholding and connected Okubo–Weiss methods described in Sect.2.3 to one of the vorticity fields from Fig. 2, with $T = 0.14$ and $k = 0.023$, respectively. These parameter values were chosen to maximize the amount of circulation enclosed by each large-scale vortex without causing "bridges" to form between adjacent ones, and to capture comparable amounts of circulation, so that the two methods could be evaluated against one another. The specifics of the tuning process, and the associated issues, are discussed below. Both methods successfully identified regions of the flow where the vorticity is high, though their determination of the boundaries are somewhat different. The dip in vorticity on the right side of vortex in the middle of the image, for instance, was resolved as an indentation by connected vorticity thresholding but as an enclosed hole by connected Okubo–Weiss, and the boundary of lowermost vortex in the field is far larger in Fig. 3b. The total positive and negative circulations of the measured flow, calculated from the vorticity data shown in Fig. 3a, were 0.0563 and −0.0584, respectively, in normalized units.[4] The connected components in Fig. 3b and c capture only part of that amount, of course, since they only include the high-vorticity regions. The large-scale vortex structures found by connected vorticity thresholding capture 66% of the overall positive circulation and 72% of the negative circulation; those extracted by connected Okubo–Weiss capture 67 and 71%, respectively.

The thresholds T and k, which these methods use to define the edge of a vortex, will obviously affect these numbers. The associated issues are standard in all vortex extraction methods that use these kinds of thresholds: lower values for either parameter increase the size of the connected components and the amount of the total circulation that they capture, but may also create bridges between neighboring structures. The mathematics behind the tuning process is slightly different in the two cases. Both are ratiometric, but they involve different scales: simple vorticity magnitude in the case of T and the difference between the magnitudes of the rate of rotation and the rate of strain in the case of k. Changing the T parameter in connected vorticity thresholding and the k parameter in connected Okubo–Weiss will thus have slightly different effects because the latter discriminates between different kinds of vorticity-bearing structures while the former simply reacts to the magnitude of ω. In the case of Fig. 3, this led to the formation of bridges between like-signed vortices in (b) and *unlike*-signed vortices in (c). In both cases, the number of large-scale structures reached an obvious plateau as the corresponding parameter was decreased, which was then followed by an abrupt increase. The reasons for this are straightforward. At large values of the tuning parameter, the number of structures satisfying the criteria dropped to zero. At very low values, noise in the low values of the vorticity field rapidly increased the number of structures identified. For a relatively smooth field that is dominated by structures large compared to the grid spacing, a band of values will exist between these extremes where the number of

[4] Lengths and velocities are scaled as described at the end of Sect. 2.1.

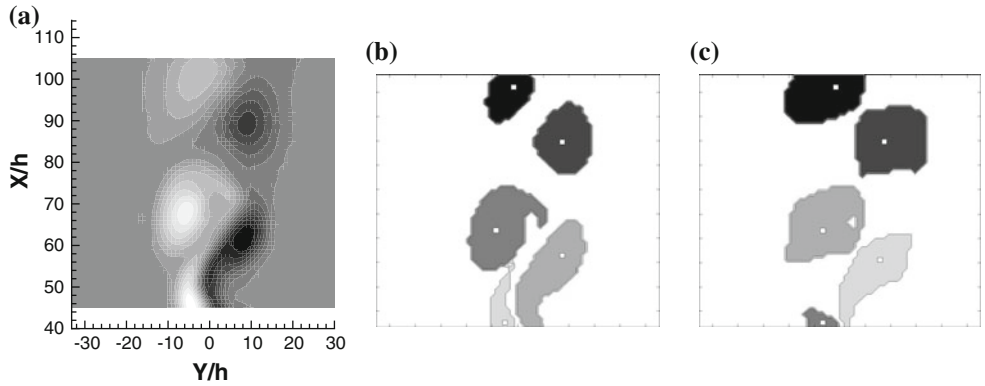


Fig. 3 Vortex extraction: **a** reproduces the vorticity field from Fig. 2g. Images **b** and **c** show the large-scale vortex structures found in that data set by the connected vorticity thresholding method of Sect. 2.3.1 with $T = 0.14$ and the connected Okubo–Weiss method of Sect. 2.3.2 with $k = 0.023$, respectively. The color scale in **b** and **c** distinguishes different vortices; the small white square inside each indicates the associated vorticity maximum

structures varies slowly. One can use this feature of the curve to choose values for the tuning parameters.

3.2 Discretization

As described in Sect. 2.3.3, three different strategies were evaluated for distributing discrete vortices to match a given snapshot of a vorticity field: a fine-grained method (i) that placed discrete vortices of appropriate strength at every grid point, a coarse method (ii) that used a single discrete vortex to model each large-scale vortex, and a parametrized method (iii) that distributed a variable number of uniform-strength discrete vortices across each large-scale vortex. Each strategy was evaluated by comparing the instantaneous velocity field induced by the resulting vortex configuration to the PIV data from which the original vorticity field was computed.

Figure 4b shows the velocity field for strategy (i) applied to the vorticity field of Fig. 2g, with the true velocity field reproduced alongside for easy comparison. Recall that strategy (i) calculates the circulation in each grid cell, placing a discrete vortex with the corresponding strength at every grid point. The instantaneous velocity field induced by this configuration of vortices was calculated using the blob model, as described in Sect. 2.4. The core-radius (r_c) value was optimized by minimizing the mean-squared error between measured and reproduced fields, as shown in Fig. 5. The optimal value for this discretization was found to be on the order of the grid spacing ($r_c = 0.6h$), a result that is consistent with the underlying mathematics. To the eye, the induced field in Fig. 4b is a fairly good match to the measured flow in part (a). The vortices and the large-scale mean flow of the jet are similar, though there are some discrepancies within three grid rows of the boundaries because of edge effects.[5] These edge effects cause a 1% loss of circulation between parts (a) and (b) of Fig. 4. The mean-squared error between the two fields is 0.0177 m^2/s^2, which is 1.3% of the square of the maximum velocity in the measured field.

Figure 6 shows the velocity fields produced by the coarse-grained discretization—strategy (ii)—of the connected components in Fig. 3. Part (a) of the figure again shows the PIV data for comparison. Parts (b) and (c) show the velocity fields that are induced by single discrete vortices placed at the high-$|\omega|$ seed point of each connected component found by vorticity thresholding and Okubo–Weiss in that PIV data, respectively. These fields were again computed via the blob model with the vortex core radius value that minimized the mean-squared error between measured and induced fields: $r_c = 6.4h$ for connected vorticity thresholding and $r_c = 6.5h$ for connected Okubo–Weiss. If the extraction and model were perfect, of course, the fields in Figs. 6b and c would be identical to the original velocity field in (a). While both configurations do indeed capture the basic structure of the flow, the induced fields show some error in the upstream regions, where the original vortices are elongated and thus less well described by the blob model. This disparity is not surprising; reducing the vorticity field of a real-world flow to five discrete vortices is a radical approximation—and one that completely distorts the vorticity distribution by forcing the cores to be circular. The mean-squared errors between the induced fields of Fig. 6b and c and the measured field of (a) were 3.5 and 3.3%, respectively. Placing the discrete vortex at the centroid of each large-scale vortex structure, as

[5] i.e., the effects of the vortex structures that are outside the field of observation, which factor into the velocities of Fig. 4a but not those of Fig. 4b.

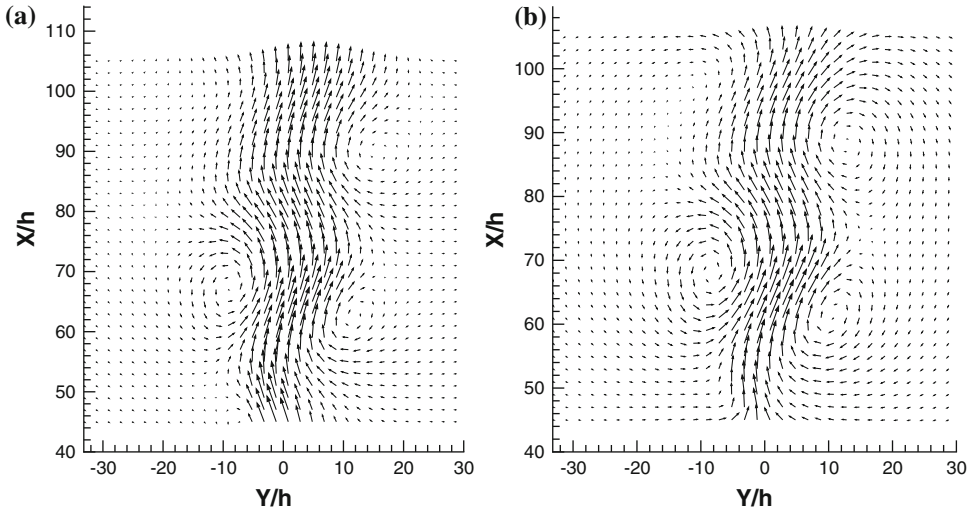


Fig. 4 Evaluation of fine-grained vortex discretization: flow fields. The image in **a** reproduces the velocity field from Fig. 2c. The image in **b** shows the velocity field induced by a configuration of blob vortices with one at every grid point of Fig. 2g and a vortex core radius $r_c = 0.6h$

shown in Fig. 7—rather than the location of peak vorticity—raised the error to 3.7 and 3.4%, respectively. All of these error values are of course significantly larger than the error in the fine-grained discretization of Fig. 4b, where the field was modeled by almost 800 times as many vortices.

Increasing the fidelity of the modeling approximation—using better models of each vortex element, increasing the number of discrete vortices, and/or tailoring their strengths and positions to the flow—will generally improve its accuracy, but at a computational cost that can be prohibitive in a data assimilation application, where correction frequency and hence data-processing speed are critical. The parametrized approach described as strategy (iii) in Sect. 2.3.3—is designed to allow the discretization process to be tuned to balance this trade-off. Figure 8 shows how this strategy discretizes the large-scale vortices of Figs. 3b and c with a q value of two (i.e., a vortex strength quantum equal to half of the strength of the weakest vortex in the field). The numbers of discrete vortices used to discretize each connected component reflect the corresponding vortex strengths: {2, 4, 7, 8, 2} small vortices in Fig. 8b, from top to bottom, and {6, 10, 14, 14, 2} in (c). To evaluate these results, the induced velocity fields were computed for the vortex configurations of Fig. 8b and c, again using the blob model with r_c chosen to minimize the mean-squared error in each case, then compared to the PIV velocity field snapshot. Visually, the induced fields are indistinguishable from Fig. 6b and c and so are not shown. The mean-squared differences between these fields and the measured field were 3.6 and 3.3%, respectively. $r_c = 6h$ minimized the mean-squared error in both cases; the effects of changing r_c and q are shown in Fig. 9. Overall, the Okubo–Weiss results are slightly better, and there is some improvement in both methods with increasing q, though not as much as one might expect. For Okubo–Weiss, the

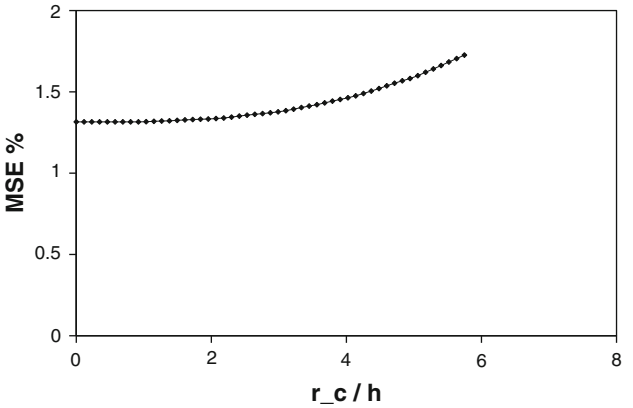


Fig. 5 Mean-squared error in induced velocity field of fine-grained vortex discretization as a function of r_c, expressed as a percentage of the square of the maximum velocity

$q = 2$ error values are comparable to those of the coarse discretization, even though the former involved more than nine times as many discrete vortices (46 instead of five). There was some further improvement with increasing q; at $q = 6$, with 141 discrete vortices, the error was 3.2%, compared to 3.3% for the coarse discretization's five vortices. For vorticity thresholding, the error value for the $q = 2$ discretization of the structures found by vorticity thresholding is actually slightly *worse* than the coarse discretization results of Fig. 6b, even though the former models the flow with almost five times as many vortices (23 instead of five). This is likely because the additional vortices are clustered around the peak vorticity location and primarily within the core radius, which effectively increases the core radius. The error improves as q is raised, eventually equaling that of the coarse discretization at $q = 6$, with 69 discrete vortices.

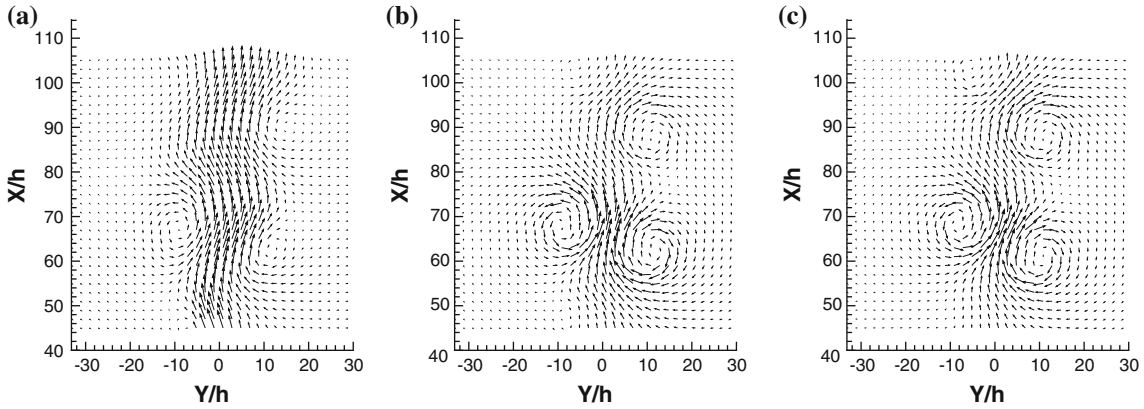


Fig. 6 Evaluation of coarse-grained vortex discretization: flow fields. The image in **a** reproduces the velocity field from Fig. 2c. Images **b** and **c** are the velocity fields induced by coarse-grained discretizations of Figs. 3b, c, respectively: i.e., a single equivalent-strength blob vortex placed in each connected component. r_c was chosen to minimize the mean-squared error for each case, yielding $r_c = 6.4h$ in (**b**) and $r_c = 6.5h$ in **c**

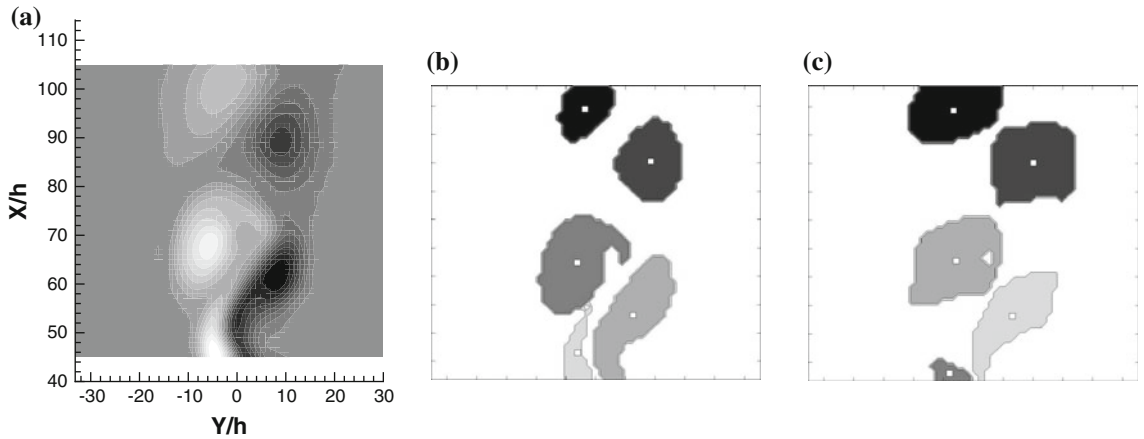


Fig. 7 The locations of the centroids of the large-scale vortex structures in Fig. 3a, which is reproduced here as part **a** for easy comparison. **b** Vorticity thresholding, **c** Okubo–Weiss

3.3 Another case

Data assimilation strategies perform corrections repeatedly, using a series of measurements of the state of the target system, so it is important to move beyond studying a single snapshot of the flow field when assessing data-processing techniques for these purposes. To this end, the analyses presented in the previous two sections were repeated on one of the other flow fields from Fig. 2. The maximum velocity in this field—Fig. 2a—was 1.22 m/s. The tuning procedure described in Sect. 3.1 yielded $T = 0.127$ and $k = 0.015$—somewhat higher than the values for the field analyzed in the previous sections (0.14 and 0.023, respectively). At these values, both extraction strategies found five large-scale vortices. The structures found by connected vorticity thresholding captured 70 and 62% of the total positive and negative circulation of the field; those found by connected Okubo–Weiss captured 72 and 61%, respectively. These vortex structures are shown in the top row of Fig. 10. For both methods, the optimal r_c values for the coarse discretizations were $6h$; the mean-squared errors of the fields induced by these discretizations were 3.3 and 3.2% for connected vorticity thresholding and connected Okubo–Weiss, respectively. These errors were roughly 2.5 times as large as the mean-squared error of the field induced by the discretization of strategy (i), shown in the bottom row of Fig. 10, which uses roughly 800 times as many vortices. This is consistent with the results in the previous sections. As before, the optimal core-radius value for this fine-grained discretization was found to be on the order of the grid spacing. The results for the parametrized discretization were also similar to those in the previous sections. Discretizing the large-scale structures found by connected vorticity thresholding produced mean-squared errors ranging from 3.2% at $q = 2$ to 3.1% at $q = 6$. For Okubo–Weiss with $q = 2$, the error was 3.0%; increasing q

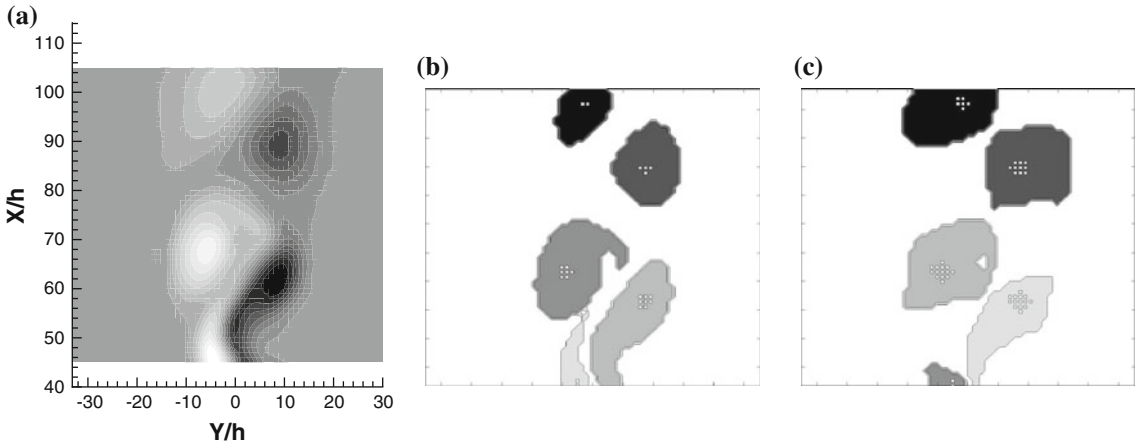


Fig. 8 Parametrized vortex discretization: the image in **a** reproduces the vorticity field from Fig. 2g. The grey-scale regions in **b** and **c** show the vortices found in that data set by connected vorticity thresholding and connected Okubo–Weiss, respectively, as in Fig. 3. The small white squares inside each of these connected components indicate where the discretization strategy (iii) in Sect. 2.3.3 placed discrete vortices: 23 in **b** and 46 in **c**. The strength quantum for the discretization in this Figure was half of the strength of the weakest vortex in each case (i.e., $q = 2$)

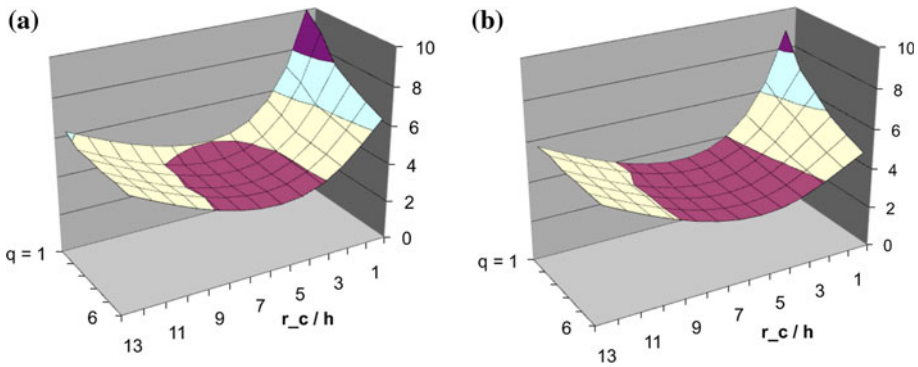


Fig. 9 Effects of quantization parameter q and vortex core radius r_c upon mean-squared error of results, expressed in %: **a** connected vorticity thresholding, **b** connected Okubo–Weiss

beyond 2 produced little improvement, and even worsened the error slightly, to 3.1% at $q = 6$. The optimal r_c value was $6h$ for all discretizations of both methods.

3.4 Discussion

Some of the results in the previous sections are consistent with previous studies, while others were unexpected. The extraction methods fall into the previous category. In terms of the accuracy of the reconstructed velocity fields, results for connected Okubo–Weiss are 5–10% better than for connected vorticity thresholding—in either coarse or parametrized discretizations. This is not surprising, as Okubo–Weiss is more sophisticated. But it is also computationally more expensive: the additional computation of the gradient components for each grid point increases the runtime of the algorithm by a factor of four. The computational cost of the connected-component algorithm, in the worst case, is the same as that of any other algorithm that groups points into large-scale structures, since both may have to "touch" every grid point. In practice, however, it is far faster, since it follows the natural flow geometry from point to point. See Ross (2008) for a full complexity analysis. This accuracy/complexity trade-off, which becomes very important in the context of data assimilation, is discussed in more depth below.

Comparing individual discretization strategies is somewhat surprising, at first glance, however, in that using a finer-grained discretization (viz., higher q) does not always improve the accuracy of the reconstructed field. The series of experiments reported in the previous section used between five and 141 vortices to discretize the five large-scale vortex structures, exploring a range of $1 < q < 6$ for the parametrized discretization strategy (iii) and comparing it to the coarse strategy (ii). The mean-squared errors of the latter were similar to those of the former: generally slightly worse and sometimes slightly better. For the field of Fig. 2a, the 46-vortex $q = 2$ discretization was comparable to the five-vortex coarse discretization; for Fig. 2g, the $q = 2$ discretization was actually *worse* than the coarse one, even though it involved eight times as many vortices. For all large-scale vortex structures, the error of the

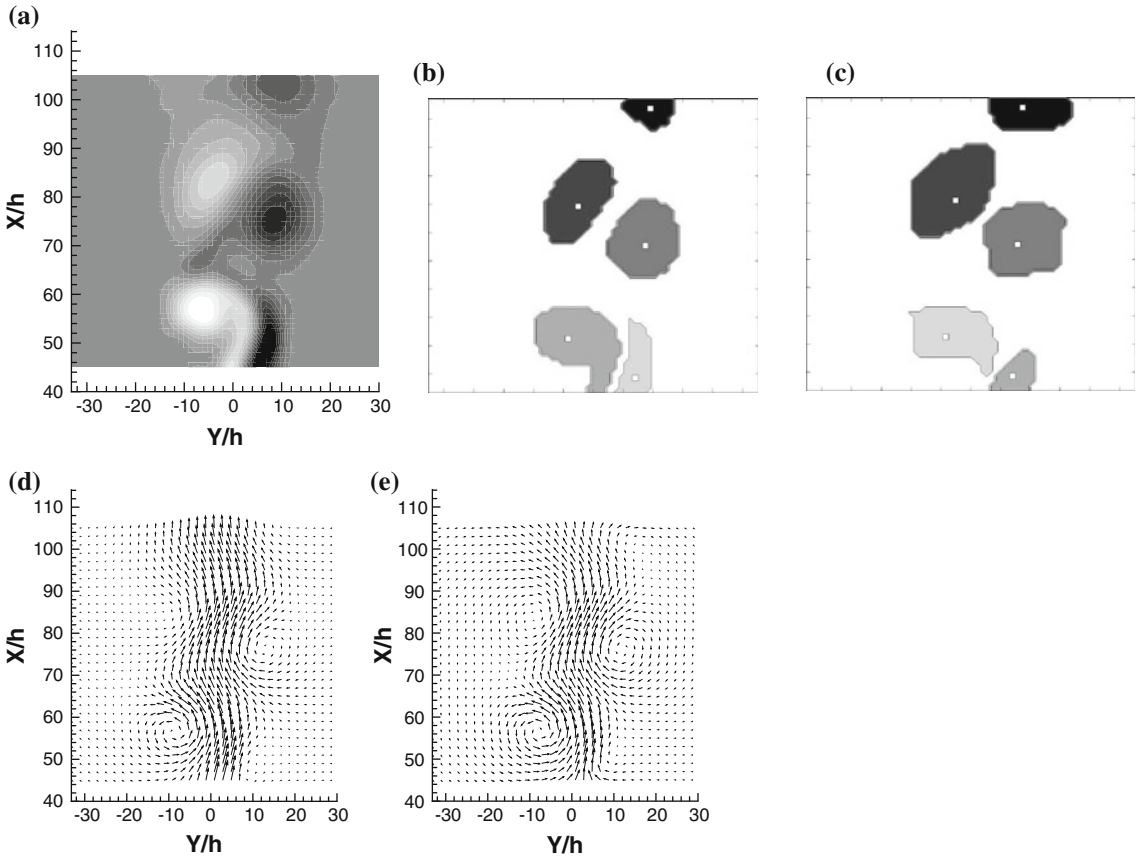


Fig. 10 Analysis of Fig. 2a. *Top row*: **a** original vorticity field and large-scale vortex structures found in that data by **b** connected vorticity thresholding with $T = 0.127$ and **c** connected Okubo–Weiss with $k = 0.015$. *Bottom row*: a comparison of the original velocity field (**d**) and the velocity field induced by a configuration of blob vortices (rc = 0.6h) with one at every grid point of that field **e**

parametrized discretization did shrink somewhat as the strength quantum of the discrete vortices was lowered—except for Fig. 10c, where the error rose from 3.0 to 3.1% as q changed from 2 to 6 (resp., 26–77 vortices). Recall that increasing the q parameter in strategy (iii) reduces the strength of the discrete vortices that are used to model each large-scale vortex. Clustering additional vortices of uniform value around the local vorticity peaks, however, serves only to diffuse the impact of the vortex on the velocity field—in effect, merely increasing r_c. Note that strategy (iii) does not reduce to the other discretization strategies in the limit of large or small q. The coarse discretization strategy models each large-scale vortex with a single discrete one; strategy (iii) with $q = 1$ models the *smallest* large-scale vortex with one discrete vortex—and more than one to model the others. Increasing q increases the number of discrete vortices used to model each large-scale vortex; the fine-grained discretization strategy, in contrast, always places one discrete vortex *at each grid point*, with strength computed to match the local vorticity. Because that strategy distributes vortices throughout the flow, rather than clustering them near the peaks of the large-scale vortices, and because it tailors their strengths to the data in each region, it does provide increased resolution—particularly in terms of capturing the large-scale flow and the less-regular vortex structures.

Mean-squared error between two static velocity fields is not the only meaningful metric in this application. A finer discretization will, in general, allow a model to track the deformation of the cores more effectively *over time*, though an n-fold increase in the number of vortices will cause a $3n$-fold increase in the number of state variables in the 2D model. Evaluating this accuracy/complexity trade-off is far from straightforward in a data assimilation application, where flows are extremely complex and their future evolution is modeled by a hybrid process that combines time-stepping of the model equations and periodic correction using the measured data. The standard discretization approach in the vortex-methods literature is to place one discrete vortex at each grid point in the vorticity field and make the grid as dense as feasible. This represents one end of the modeling spectrum, where the discretization procedure is simple, the grain of the approximation is very fine, and the model contains a large number of state variables. If

one is interested in speed, however, this may not be desirable. The vortices in the fine-grained discretization of Fig. 4b require roughly twelve thousand state variables to describe[6], which make it run on the order of a hundred times more slowly than a model that uses the parametrized discretizations of Fig. 8 and almost 2,500 times more slowly than a model that uses the coarse-grained discretization of Fig. 6.

Recall, though, that running the model is only part of the computational cost of the larger process of data assimilation. Indeed, the complexity of these algorithms is typically dominated by the cost of processing the data that they use to correct their models. In traditional data assimilation approaches, where corrections are applied periodically, one can simply add up the costs of processing the data (described in the first paragraph of this section) and stepping the model, which is linear in the number of state variables. The cost of "correcting" with bad data is potentially huge, however, as it can cause the models to diverge beyond repair, requiring a restart. Cutting-edge data assimilation approaches adapt the correction timing to the combined, evolving dynamics of the system and model, which makes any kind of complexity analysis completely intractable. For all of these reasons, it is imperative to examine—and truly understand—the effectiveness of the methods used to process the data. It is also critical to provide tools for the people who design the algorithms that use those data, so that they can tailor the process to the problem at hand.

The problems that arise in numerical weather prediction and other fields that employ data assimilation are far more complex and much harder to measure than a planar jet in a fluid dynamics laboratory. The planar jet flow in Fig. 2 is smooth, has a limited set of length scales that do not change with time[7], and can be well measured and characterized. Complex, time-varying real-world flows pose more challenges. To begin with, the inherent lumpiness of approximating a sheet with a collection of discrete vortices would be an issue. In a flow that has a broad range of length scales and non-isolated structures, tuning k and T would be extremely difficult, and that process would have to be repeated as the flow changed. This process could be automated, following the approach outlined at the end of Sect. 3.1, and perhaps using wavelets to estimate the scale (though our preliminary studies show that the computational cost of the wavelet analysis may be too high to make that practical). And while no data-processing method can account for scales that are below the measurement resolution, the data are reasonably well resolved for the domain examined here, where the flow is laminar and the velocities vary smoothly. The methods described here will likely not scale to fully turbulent flows—though they may apply to their smaller scales. Of course, none of these issues and caveats are specific to the work described here; they are general issues for any vortex extraction method when the flow is turbulent.

[6] Discrete vortex positions and strengths at 61 × 65 grid points.

[7] though the individual vortices do change as they enter, move through, and exit the field of view.

4 Conclusion

The vortex extraction and discretization techniques described in this paper are designed to support data assimilation algorithms, which use information from sensors to correct the state variables of a numerical solver *while it runs*. As such, speed is critical, careful attention to temporal convergence of the model is not necessary, and there is no need to distinguish between vortices and other kinds of vorticity-bearing structures like shear layers. Rather, this application demands a fast, easily adaptable extraction/discretization strategy that models a snapshot of the flow with as few or as many discrete vortices as are demanded by the situation at hand. The methods described here accomplish this by augmenting classic vortex extraction methods with a simple computational topology technique. Specifically, they define a single vortex by searching for a connected component of high vorticity. Each of these large-scale structures is then modeled using a collection of uniform-strength discrete vortices distributed around its vorticity peak.

The findings in Sects. 3.1–3.3 show that the connected Okubo–Weiss method is slightly more accurate than the connected vorticity thresholding in extracting the large-scale vortex structures in a flow, at least in the fields considered here. In view of the greater computational complexity of Okubo–Weiss, then, this suggests that vorticity thresholding may be preferable for the purposes of data assimilation. The discretization results were somewhat surprising, in that the mean-squared errors of the coarse versions were similar to those of the corresponding finer-grained ones: generally slightly worse and sometimes slightly better. In view of the additional terms that each discrete vortex adds to the model, this suggests that the coarse discretization strategy may be better suited for data assimilation applications, as it provides the best combination of low error and low computational complexity.

In general, of course, finer-grained discretizations produce more-realistic vorticity distributions, but the results here are not consistent with that generalization. This is likely a consequence of the clustered nature of the distribution used in strategy (iii), and the fixed strength of its constituent vortices. The obvious next steps in the future-work plan for this project are to distribute those discrete

vortices more realistically across the large-scale vortex structures and/or use non-uniform strengths. This would add computational complexity, though, and without full testing in the context of a data assimilation system, it is difficult to estimate the impact of these modifications. In these systems, snapshots of the real-world conditions are used to correct the solver periodically—preferably as often and as quickly as possible. The measurement accuracy and correction timing dominate the overall data assimilation process, which makes it critical for the preprocessing step to balance speed and accuracy. The methods described here are designed to do exactly that, in an explicit way, via the thresholding and discretization parameters described in Sects. 2.3.1–2.3.3. These tuning parameters allow—and require—a user of these methods to optimize between accuracy and computational cost for a given application.

The dynamics of data assimilation—how data processing and correction dynamics interact in different models—are complex and poorly understood. Meteorologists have known for years (e.g., Harms et al. 1992) that haphazard incorporation of sensor data into a running simulator can degrade accuracy, excite unfriendly numerical dynamics, and even destroy convergence, all of which has dire implications for the model and for anything that relies upon it. Because oceanic and atmospheric systems are so complex, however, there has been little controlled study of these effects in the data assimilation literature. The approach described here—comparatively simple system, laboratory setting, advanced sensor technology, and reduced-order model—is designed to support exactly that kind of study, and the specific goal of this paper was to explore the associated data-processing issues. The ultimate objectives of this line of work are not only to learn how to dynamically tailor the acquisition, processing, and use of correction data in the specific case of the 2D point-vortex model of the planar jet, but to derive a more-general understanding of the dynamics of data assimilation. Of course, full velocity fields are almost never available for the kinds of real-world flows that dominate current data assimilation applications. One can, however, isolate vortices in oceanic and atmospheric flows using feature-recognition techniques on remote-sensing data, and any technique that effectively incorporated reduced-order models into data assimilation would transform the field. And data assimilation is not the only potential application here; these results are of potential interest in any situation where one must balance accuracy and expense while extracting vortices from a snapshot of a flow field.

Acknowledgments the authors wish to thank Jeffrey Anderson, Matthew Culbreth, Nathan Farrell, and Mark Rast for their input to this project.

References

Adrian R, Christiensen K, Liu Z (2000) Analysis and interpretation of instantaneous turbulent velocity fields. Exp Fluids 29:275–290

Anderson JL (2010) Personal Communication

Banks D, Singer B (1995) A predictor-corrector technique for visualizing unsteady flow. IEEE Trans Vis Comput Graph 1(2):151–163

Barba L (2007) Spectral-like accuracy in space of a meshless vortex method. In: Leitão V (ed) Advances in meshfree techniques. Springer

Barba L, Leonard A, Allen C (2005) Advances in viscous vortex methods—meshless spatial adaption based on radial basis function interpolation. Int J Numer Methods Fluids 47:387–421

Camussi R (2002) Coherent structure identification from wavelet analysis of particle image velocimetry data. Exp Fluids 32:76–86

Chong M, Perry A, Cantwell B (1990) A general classification of three-dimensional flow fields. Phys Fluids 2:765–777

Chorin A (1973) Numerical study of slightly viscous flow. J Fluid Mech 57:785–796

Cottet G-H, Koumoutsakos P (2000) Vortex methods: theory and practice. Cambridge University Press, Cambridge

Farge M, Schneider K, Kevlahan N (1999) Non-Gaussianity and coherent vortex simulation for two-dimensional turbulence using an adaptive orthogonal wavelet basis. Phys Fluids 11(8):2187–2201

Farge M, Schneider K, Pellegrino G, Wray A, Rogallo R (2003) Coherent vortex extraction in three-dimensional homogeneous turbulence: comparison between CVS-wavelet and POD-Fourier decompositions. Phys Fluids 15(10):2886–2896

Farrell N (2008) PIV analysis of forcing a planar jet using a loudspeaker. Technical Report CU-CS (Department of Computer Science) 1043-08, University of Colorado

Gustafson K, Sethian J (1991) Vortex methods and vortex motion. Society for Industrial & Applied Mathematics, Philadelphia

Hald O (1991) Convergence of vortex methods. In: Gustafson K, Sethian J (eds) Vortex methods and vortex motion. Society for Industrial & Applied Mathematics, Philadelphia

Haller G (2005) An objective definition of a vortex. J Fluid Mech 525:1–26

Harms D, Raman S, Madala R (1992) An examination of four-dimensional data-assimilation techniques for numerical weather prediction. Bull AMS 73:425–440

Hua B, Klein P (1998) An exact criterion for the stirring properties of nearly two-dimensional turbulence. Physica D 113:98–110

Hua BL, McWilliams J, Klein P (1998) Lagrangian accelerations in geostrophic turbulence. J Fluid Mech 366:87–108

Hunt J, Wray A, Moin P (1988) Eddies, streams, and convergence zones in turbulent flows. In: Proceedings of the summer program, vol CTR-S88. Stanford University Center for Turbulence Research Report, pp 193–208

Ide K, Ghil M (1997) Extended Kalman filtering for vortex systems. Part I: methodology and point vortices. Dyn Atmos Oceans 27:301–332

Ide K, Kuznetsov L, Jones C (2002) Lagrangian data assimilation for point vortex systems. J Turb 3:1–7

Jeong J, Hussain F (1995) On the identification of a vortex. J Fluid Mech 285:69–94

Joseph R, Viglione S, Wolf H (1964) Cloud pattern recognition. In: Proceedings of the 19th ACM national conference. pp 42.301–42.3017

Lugt H (1979) The dilemma of defining a vortex. In: Muller U, Roesner K, Schmidt B (eds) Recent developments in theoretical and experimental fluid mechanics. Springer, London, pp 309–321

Okubo A (1970) Horizontal dispersion of floatable trajectories in the vicinity of velocity singularities such as convergencies. Deep Sea Res 17:445–454

Palacios A, Armbruster D, Kostelich E, Stone E (1996) Analyzing the dynamics of cellular flames. Physica D 96(1–4):132–161

Pemberton R, Turnock S, Dodd T, Rogers E (2002) A novel method for identifying vortical structures. J Fluids Struc 16:1051–1057

Preisendorfer R (1988) Principal Component analysis in meteorology and oceanography. Elsevier, NY

Raffel M, Willert C, Kompenhans J (1998) Particle image velocimetry: a practical guide. Springer, NY

Robins V, Abernethy J, Rooney N, Bradley E (2004) Topology and intelligent data analysis. Intell Data Anal 8:505–515

Robins V, Meiss J, Bradley E (1998) Computing connectedness: an exercise in computational topology. Nonlinearity 11:913–922

Robins V, Rooney N, Bradley E (2004) Topology-based signal separation. Chaos 14:305–316

Ross N (2008) Understanding the dynamics of point-vortex data assimilation. PhD thesis, University of Colorado

Scarano F, Benocci C, Riethmuller M (1999) Pattern recognition analysis of the turbulent flow past a backward facing step. Phys Fluids 11(12):3808–3818

Segur H (1998) Evolution of a tracer gradient in an incompressible, two-dimensional flow. In: IUTAM symposium on developments in geophysical turbulence

Seigel A, Weiss J (1997) A wavelet-packet census algorithm for calculating vortex statistics. Phys Fluids 9(7):1988–1999

Shadden S, Lekien F, Marsden J (2005) Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. Physica D 212:271–304

Vollmers H (2001) Detection of vortices and quantitative evaluation of their main parameters from experimental velocity data. Measure Sci Technol 12:1199–1207

Weiss J (1991) The dynamics of enstrophy transfer in 2-dimensional hydrodynamics. Physica D 48:273–294