Dynamics-Informed Data Assimilation in a Qualitative Fluids Model

Natalie Ross and **Elizabeth Bradley**
Department of Computer Science
University of Colorado
Boulder, CO 80309

Jean Hertzberg
Department of Mechanical Engineering
University of Colorado
Boulder, CO 80309

Abstract

Fast and accurate numerical models are critical for the modelling, prediction, and control of fluid flows. Direct numerical simulation (DNS) methods, though accurate, are often too slow for these purposes. So-called *reduced-order models* are faster because they use fewer state variables to approximate the flow physics. Different tactics are used for this dimensional reduction. Some approaches simply coarsen the numerical grain of the approximation. Others take a more-qualitative approach, decomposing the flow into abstract features—coherent structures like vortices, for instance—and modelling the dynamics of those features. Regardless of the tactics involved, the inherent approximations make reduced-order models inaccurate. The premise of this paper is that periodically correcting such a model with observations of the fluid—a process known as data assimilation—can produce a "data-adaptive" model that is both fast *and* accurate. This idea has been explored in depth by the numerical weather prediction community in the context of DNS models. The goal of this paper is to explore data assimilation in the context of a model that treats a fluid flow as a collection of vortices. There are two challenges in assimilating data into such a model: correction dynamics and computational cost. The strategy described here solves both of those problems using knowledge about the flow dynamics to intelligently select when and where to apply the correction.

Introduction

Due to the complexity and sensitivity of fluid flows, numerical models that accurately track their evolution are currently too slow for many applications. The ability to model these flows accurately and quickly is of great practical importance, however, as they are common in natural and man-made systems. Traditional approaches to numerical modelling, reviewed in the next section, use so-called *direct numerical schemes* to solve the complex partial differential equations that govern flow dynamics. While these techniques can be highly accurate, they are also very slow because they model the dynamics at all points on a fine mesh. Our approach to solving this problem is to use a more-abstract model that tracks only the coherent structures in the flow and self-corrects using physical data measured from that flow. This combination of qualitative modelling and quantitative observations results in a *data-adaptive* simulation that is both fast and accurate.

The point-vortex model tracks only the vortices in a flow, ignoring all other dynamics. In comparison to direct numerical simulation techniques, this is a *reduced-order model*: its state variables are the positions and strengths of the vortices rather than gridded velocity fields. This abstraction of the flow—as a collection of interacting coherent structures, rather than a physical continuum of velocities and pressures, or a gridded approximation thereof—has a variety of advantages. Reasoning in terms of the vortices in a flow is much simpler, and hence numerical models that encapsulate such reasoning are significantly faster. And, this abstract reasoning is justified from a physical standpoint, as it is the coherent regions in a fluid flow that are responsible for the qualitative flow behavior(Roshko 1976). Vortices, in particular, are good coherent structures to track, as fluid physics enables us to compute the velocity at any point in the flow if we know the positions and strengths of its vortices. Vortices may also be useful in helping humans to reason about turbulent flows (Yip 1995). The mathematical details of the point-vortex model are provided in the next section.

There is a drawback to using reduced-order models: the approximations that make them fast also introduce inaccuracies. Our solution to this problem is to correct the model variables on the fly using data from the target system. This process, known as data assimilation, was developed by meteorologists in the 1950s for integrating weather observations into numerical prediction models. The thesis of our work is that correcting a reduced-order model—one that is based upon qualitative features of the flow—with quantitative information is a good way to improve accuracy, and that the resulting data-adaptive reduced-order model can be both fast *and* accurate. Data assimilation presents its own set of challenges, however, as it can potentially destabilize an otherwise-stable numerical scheme. And, the computational costs of a naive assimilation strategy could negate the increase in speed obtained by using a reduced-order model. The data-assimilation community has explored these issues in great depth in regard to DNS models, but there has been almost no work on how to use data to correct *reduced-order* models, nor has anyone investigated the use of real data in this context, let alone addressed the computational cost issue in any systematic way.

In this paper, we present an intelligent data-assimilation strategy that exploits knowledge about the flow dynamics to correct the point-vortex model only when necessary. This is in contrast to traditional data assimilation approaches, which correct the model whenever observations are available, ignore the dynamics of the underlying system, and use statistical approaches to handle noise. We believe that a thorough grasp of the dynamics of the system can be used not only to inform the assimilation process, but also to understand the scenarios in which noise—an inescapable feature of any real application—may enhance or destroy its benefits.

Many of the themes in this work are familiar ones to the QR community. We are interested in abstraction as a way to simplify simulation (e.g., (Clancy & Kuipers 1993)), but we are working with a spatiotemporally extended system whose behavior cannot be envisioned or enumerated. In contrast to (ky Ringo Ling & Steinberg 1993), we are not lumping control volumes to transform the system into an ODE, nor is our goal diagnosis, as in (Sachenbacher & Struss 2001; Struss 2002; Yan 2003). To simplify our simulations, we are using a reduced-order model that tracks the coherent structures in a fluid. The goal in the current paper, however, is not to find or understand those structures (Bailey-Kellog & Zhao 1997; Bailey-Kellog, Zhao, & Yip 1996; Ordóñez & Zhao 2000; Nishida 1993; Yip 1995; 1997), but simply to use data about them to correct a simulation of their dynamics. Like Lundell (Lundell 1994; 1995; 1996), we are building a qualitative model of a physical field; the form of the model is very different, though, and we are using data assimilation to improve its accuracy. Like Zhao (Zhao 1994), we are exploiting knowledge about dynamical systems to improve simulations; our application area, however, is fluid dynamics as opposed to ordinary differential equations, and our goal is to correct the simulation not understand the state space. Lastly, one of our fundamental issues is the integration of qualitative and quantitative information, which is a prevalent research issue in the QR literature. In the next few sections, we describe the qualitative model we that are working with and explain how data assimilation can be used to incorporate quantitative observations.

The Point-Vortex Method

Real-world fluids problems do not admit analytical solutions, so one has to model them numerically, and their inherent spatiotemporal complexity makes this very hard. The traditional solution to this, termed direct numerical simulation or DNS, involves discretizing the flow quantities using finite-order approximations of time and space. To get the flow details right in face of this discretization, the grids involved may need to be very fine, which translates to an extremely large state vector in a simulation of a complicated flow. The algorithmic methods used in many codes to address this issue—e.g., sparse matrix solvers—often have sensitive numerical dynamics, making it hard to get them to converge. For all of these reasons, DNS simulations of even fairly simple fluids problems require hours—or even days or weeks—of CPU time on powerful machines with large memories.

If a coarser but still meaningful representation could be used to model the dynamics of the system, the resulting numerical solver would be simpler, and hence much faster, than DNS models. There are many examples of such reduced-order representations (Canuto *et al.* 1988; Berkooz, Holmes, & Lumley 1993; Farge, Schneider, & Kevlahan 1999; Farge *et al.* 2003; Germano *et al.* 1991; Lesieur & Metais 1996; Moin 1997; Sethian 1991). Many of these are obtained via various approximations to the Navier-Stokes equations or a coarsening of the grid employed by the DNS models. It is important to note that the majority of these solutions provide abstract descriptions of the flow, but no mechanism for modelling the dynamics in terms of these descriptions. The point-vortex model(Sethian 1991), in contrast, is an abstraction that is based on the qualitative features in the flow. This method is inherently grid-free, and the state variables are meaningful flow quantities—positions and strengths of vortices—that are helpful in understanding its dynamics. The result is a huge reduction in the number of state variables required in the simulation.

The point-vortex model's dynamics are straightforward. It tracks the vortices in a flow, assuming that that flow is inviscid. Vorticity is a field vector quantity defined as the curl of the velocity; it represents the angular momentum of the fluid. A vortex is a local peak or concentration in the vorticity; circulation is the integral of vorticity over an area. In the *point*-vortex model, all vorticity is idealized as being contained at specific points, which are assumed to move with the flow field. As mentioned above, state variables in the point-vortex model are the positions (x, y) and strengths Γ of these idealized point vortices. This model's dynamics are the fluid-mechanical analog of point masses evolving under the mutual interaction of Newtonian gravity: a vortex is treated as generating a swirling velocity field around itself, and other entities—vortices, passive tracer particles—move or "advect" with that velocity. The magnitude of the induced velocity falls off as $1/r^2$ with the distance r from the corresponding vortex core. The point-vortex equations use superposition to combine the effects of multiple point vortices. In schematic form, the equations for the evolution of the state of the ith point vortex are:

$$\begin{bmatrix} \dot{\vec{X}}_i \\ \dot{\Gamma}_i \end{bmatrix} = \begin{bmatrix} \vec{f}(\Gamma_j, \vec{X}_j,\ j \neq i) \\ 0 \end{bmatrix} \quad (1)$$

where $\vec{X}_i = (x_i, y_i)^T$, the 2D position of the ith vortex, and $\vec{f}$ is a vector-valued function whose ith component computes the distances $||\vec{X}_i - \vec{X}_j||_2$ from the ith vortex to each of the j others, computes their influence at that distance (via the $1/r^2$ law, scaled by the strength Γ_j), rotates to the tangential direction, and does a vector sum of the results. One can solve the system (1) with any ordinary differential equation (ODE) solver.

The point-vortex model is highly idealized. That is what makes it fast, but idealization also introduces inaccuracy. Real vortices are not concentrated at a single point, and only higher Reynolds number flows can be treated as inviscid. More typically, vorticity is distributed throughout the flow, and it is created and destroyed as the flow evolves.

Nonetheless, the point-vortex method works remarkably well (Boyland, Stremler, & Aref 2003) if the flow is dominated by isolated regions of high vorticity, the fluid surrounding those regions is basically irrotational, and viscosity is small—assumptions that are valid for many engineering flows. There are many ways to extend the point-vortex model to handle cases where these assumptions are not valid (Airapetov 1990; Basu, Narasimha, & Prabhu 1995; Chavanis 2001; Cortelezzi, Chen, & Chang 1997; Funakoshi 1995; Huber & Alstrom 1993; Riccardi & Piva 2000). These improvements will not play roles in our research, since the solver itself is not our focus. Our goal is to figure out how to use data assimilation to improve the accuracy of the original point-vortex algorithm and present a proof-of-concept example that it works.

Data Assimilation Overview

In order to combat the small- and large-scale errors introduced by the point-vortex approximation, we correct the solver with experimental measurements of the fluid under investigation—a process known as data assimilation (see (Daley 1991; Tarantola 1987) for an overview). Figure 1 shows a schematic of the correction process, and depicts what is typically referred to as a *data assimilation cycle*. The steps in the process are as follows. In the first cycle, one must specify the initial conditions for the model integration—this is known as the *initialization problem* in the data assimilation community. Once initial conditions have been specified, the model is run for a specified time interval (cycle length) to produce a *forecast* or *background* state. This forecast step is represented by

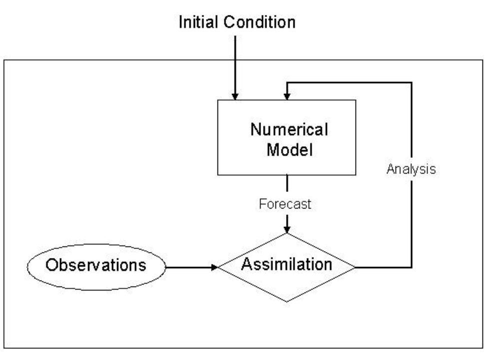


Figure 1: Data assimilation cycle. A numerical model is used to generate a forecast or "background state" from a best-guess initial condition. Data assimilation is then used to combine the background state with the available observations, each weighted according to its expected accuracy. The result is an "analysis state," which is used as the initial condition for the next assimilation cycle.

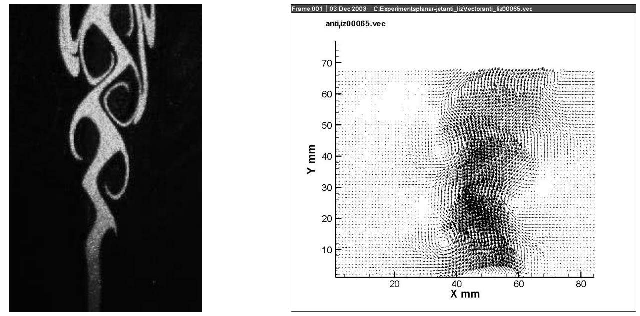
$$\mathbf{x}^{\mathrm{f}}(t_{i+1}) = M_i[\mathbf{x}^{\mathrm{f}}(t_i)]\,,$$

where M_i is the model dynamics operator and $\mathbf{x}^{\mathrm{f}}(t_i)$ is its state vector at time t_i. Finally, observations of the dynamical system are combined with this background state to produce a new model state known as the *analysis*. Depending on the analysis algorithm and the model, it may be necessary to apply initialization techniques to the analysis to ensure that it satisfies certain dynamic balance conditions. The analyzed/initialized state is then used as the initial condition to start the model forecast for the next data assimilation cycle.

This seemingly simple data assimilation cycle is rich with interesting and challenging problems. Much of the research in this field is devoted to the analysis step, i.e., determining what algorithm should be used to update the model variables, based on the available observations. One naive approach to this is to simply throw out the simulated variable values and replace them with the measured ones (where they exist). We will refer to this method as "direct replacement." There are a variety of major problems with this. To begin with, it can deliver a numerical shock to the solver, and numerical algorithms are notoriously sensitive and prone to diverge when subjected to this kind of insult. Simple control-theoretic ideas can soften this shock. The meteorology community has been using the obvious proportional control strategy to do so for thirty years (though they term it "Newtonian nudging" (Davies & Turner 1977)). Direct techniques like this are all very well if the data are plentiful, noise-free, and an exact match to the variables used in the simulator, but that is rarely the case in practice, and the bulk of the data assimilation literature is devoted to techniques for dealing with the sparseness, noise, and heterogeneity of real data.

Atmospheric and oceanic assimilation systems typically deal with these issues by working with an *observation grid* and a *model grid* and interpolating simulated and measured data back and forth between the two in order to perform the correction. This interpolation, in the face of noise and sparse data, is the main challenge of data assimilation. Early solutions used simple linear methods to solve this problem, but these did not take into account that noise and significance levels differ across data sets. The next generation of data-assimilation approaches used statistical interpolation techniques involving covariance matrices to transform between the two grids in a manner that weights different observations appropriately (Daley 1991, chapter 4). Kalman filters, a conceptually neater but much more computationally expensive way to solve this problem, came into use in this community in the 1990s, along with an *ensemble method* that uses Monte Carlo techniques to estimate the sensitivity of the model to different kinds of corrections and then tailors its actions accordingly (Anderson 2003; Evensen 1994). Another elegant approach uses techniques from variational calculus to find the model trajectory that most closely fits the observations (Dimet & Talagrand 1986). *Note that none of these techniques use knowledge about the dynamics of the system to guide the design of the correction strategy.* And, no one has thoroughly examined how the success of these statistical strategies might depend on the state of that system.

All of the aforementioned data assimilation techniques have been developed in the context of DNS simulations of large-scale atmospheric and oceanic systems; data assimilation into point-vortex models has received much less attention. Kayo Ide *et al.* (Ide & Ghil 1997; Ide, Kuznetsov, & Jones 2002) have done some interesting work in this field. This algorithm deduces the vortex positions by inverting the velocity superposition arguments that are built into the point-vortex equations and then assimilates that data into point-vortex models using Kalman filters. They have tested this strategy extensively in numerical simulations. They also developed a hybrid assimilation method that assimilates data about *both* the positions and strengths of vortices *and* the paths that tracer particles take through a flow. The basic idea is to augment the point-vortex equations (1) with a set of tracer advection equations that model the dynamics of particle movement(Ide, Kuznetsov, & Jones 2002). The key here is that the fundamental link between velocity and vorticity couples these equations, so corrections made to one will "percolate" into the other. That is, one can assimilate tracer particle data into the advection equations and the cross-coupling term will naturally carry those corrections into the point-vortex equations. Ide *et al.* have studied this approach in simulations, but it has not yet been implemented with experimental data.



(a) (b)

Figure 2: (a) A planar air jet. $\mathrm{Re} \approx 70$. Vortices are clearly visible in this photograph of the jet. Our goal is to track these coherent structures with a point-vortex model, corrected with experimental data to maintain accuracy. (b) is a sample of the raw velocity field data obtained from a particle image velocimetry (PIV) system.

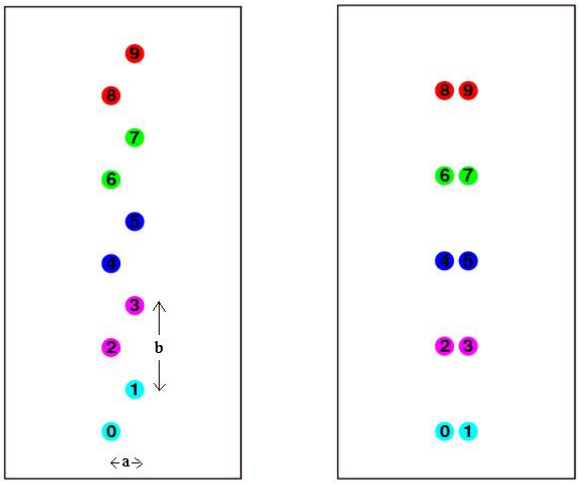
Dynamics-Informed Assimilation: Methods and Results

Our goal, and the novelty of our work, is to develop effective strategies for *timing* the assimilation of data into the point vortex model. We are using the dynamics of the system to determine when and where model corrections will have the most impact, enabling us to decide whether or not the computational cost of gathering and processing system observations is worth the effort. This paper presents a series of numerical experiments that provide a solid proof-of-concept demonstration of our strategy, which is based on the observation that solvers make mistakes when the spatial gradients of the equations that they are solving are high. Our ultimate goal, of course, is to apply this to a real-world fluid flow: a laboratory air jet that is described in the following paragraph. While this is a much simpler flow that those that geophysicists work with, it calls many of the important questions—noise, computational cost, etc.—that are ignored by all of the existing assimilation work on reduced-order models. The laboratory setting also distinguishes our work from the bulk of the data assimilation literature: it lets us effectively isolate, explore, and understand the associated research issues in a fashion that is simply not possible when one is working with a system that is as complex and hard to observe—let alone control—as the weather.

The motivating example for this work, and the testbed for the stages that will follow this paper, is a planar air jet (Peacock *et al.* 2004). Using actuators at the base of the jet, we can force the flow to assume one of its two unstable modes. A picture of the jet in its antisymmetric mode is displayed in Figure 2(a). Vortices are well-defined in both the symmetric and antisymmetric modes, which makes the forced jet a good candidate for point-vortex modelling. We also have a mechanism for gathering velocity data from this flow—particle image velocimetry (PIV). A PIV system works as follows: (1) aerosol particles injected into the fluid are illuminated by a laser light sheet, (2) a camera situated perpendicular to the light sheet takes two photographs of the flow in quick succession, and (3) the photographs are cross-correlated to determine displacements of the aerosol particles, which can be used to infer the velocity at each particle position. Figure 2(b) shows a sample velocity field of the jet, obtained via PIV. Our ultimate goal is to use this laboratory setup to investigate some of the traditional data assimilation methods described in the previous section and compare them with our dynamics-informed correction approach, which is described in the rest of this section.

As a first step toward this goal, we have devised a set of numerical simulations that comprise a meaningful test of our approach. The basic idea is common in the numerical computing community: use a fine-grained simulation as an ansatz for the "true" behavior of the system. In our case, this amounts to using a high-resolution simulation to correct a coarser one. This effectively isolates the data-assimilation research questions treated in this paper from the complications of real data, and provides a controlled scenario in which to gain experience with these techniques.

To make the ansatz as close as possible, we choose initial conditions for our model that resemble those observed in the laboratory. Figure 3 displays two initial vortex configurations that mimic the symmetric and antisymmetric modes of the jet. The vortex configuration displayed in Figure 3(a) is derived from the well-known von Karman vortex street. Von Karman proved (Lamb 1945) that two infinitely long parallel rows of vortices will remain stable if two conditions are satisfied: (1) the strength of each vortex is identical, with vortices in the left column having opposite vorticity from those in the right column and (2) the spacing between vortices satisfies $a/b = 0.281$, where a and b are labelled in Figure 3(a). Clearly, in our numerical experiments, we cannot use an infinitely long vortex street; but, even with a finite number of vortices, this arrangement will result in relatively stable dynamics. In contrast, the symmetric pattern



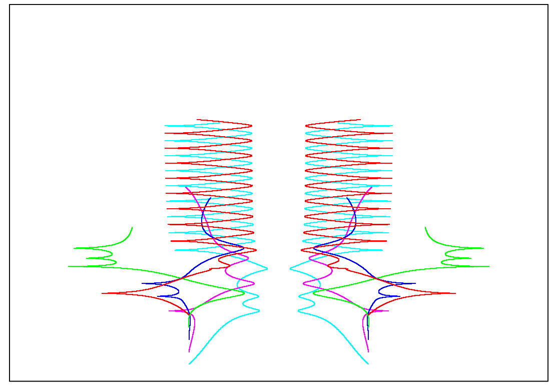
(a) (b)

Figure 3: Vortex configurations. Initial conditions in (a) are derived from the stability condition for a von Karman vortex street. Vortices are spaced 1 unit apart in the x-direction and $a/b = 1/0.281 \approx 3.6$ units apart in the y-direction. A similar vertical spacing of 3.6 units and horizontal spacing of 1 unit was used to obtain the symmetric configuration displayed in (b). In both cases, the vortices in the left column have strength -1 (counter-clockwise rotation), and those in the right column have strength 1 (clockwise rotation).

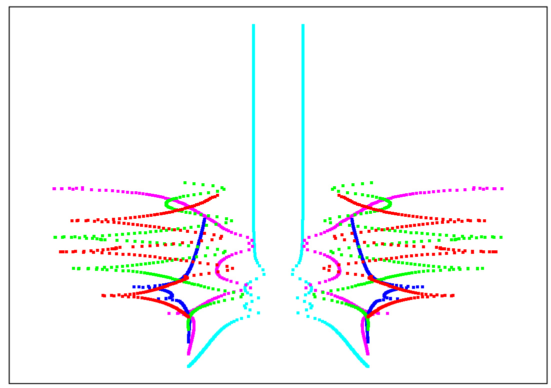
displayed in Figure 3(b), which corresponds to the symmetric mode of the jet, is highly unstable. Thus, these two vortex configurations provide two very different contexts—both of which are physically realistic—in which to study data assimilation methodologies.

Starting from these initial conditions, we first ran a high-resolution point-vortex model simulation to represent the "truth." This simulation—a 4^{th}-order Runge-Kutta solution of the point-vortex equations (1) with a small timestep—provides a relatively accurate picture of the dynamical evolution of the system, so it makes sense to use it as a stand-in for the experimental data. The vortex trajectories in this simulation are depicted in Figure 4(a). We then ran a similar simulation, shown in Figure 4(b), with a much larger time step—one large enough to cause the solution to diverge from the true value. This is a useful ansatz for what happens when a simulation diverges from reality, as floating-point error and physical noise have many of the same effects. The final step in the evaluation of our data-assimilation strategies was to use the "truth" simulation to correct the "model" one. Eventually, of course, we will be working with real data as the "truth" and a higher-resolution model of the planar air jet as the "model."

We first attempted a direct, continuous assimilation approach, simply replacing the simulated variables in the "model" run with the "true" values at various intervals. This is the standard "periodic correction" approach used in most of the data assimilation research that was reviewed in the previous section. Figure 5 shows a point-vortex simulation



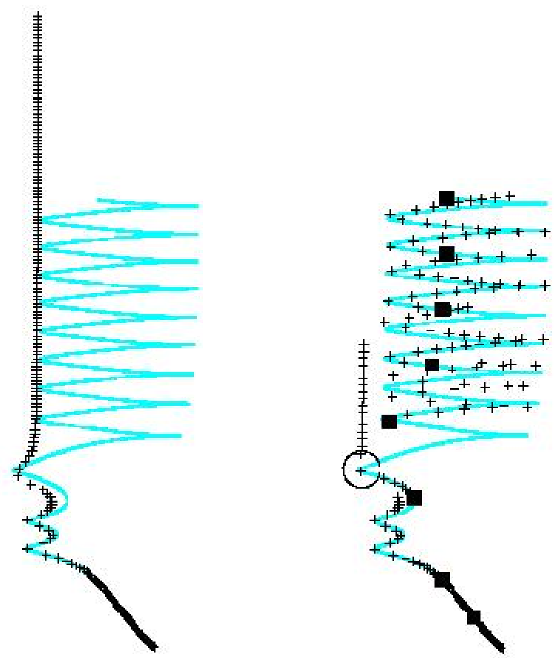
(a)



(b)

Figure 4: Full trajectories of (a) "truth" and (b) "model" simulations starting from initial conditions in Figure 3(b). Note that these plots are not to scale; we have zoomed in on the x-range to make it easier to see the interesting dynamics. (a) is a 200 second simulation of Equation 1 from the initial conditions of Figure 3(a) using RK4 and a 0.005 second timestep. (b) is a 200 second simulation with a 1 second timestep. In our numerical experiments, we use the more-accurate trajectories from (a) to correct the vortices in (b).

corrected using this technique. As outlined above, we are using a simulation with a very fine integration timestep as the reference or "true" simulation. The solid path in Figure 5(b) displays the full trajectory of one vortex in this simulation. We use observations from this reference simulation to correct a coarser timestep simulation (represented by the $+ + ++$ path in the figure). The corrections occur at the locations indicated by the black squares. Notice that toward the beginning of the simulation, the vortex is moving quite slowly, as indicated by the small distance between $+$s. The trajectory is also fairly smooth, indicating that the vortex does not encounter large velocity gradients in this region. Note that the model trajectory does not diverge from the "true" simulation, so the first two corrections applied provide very little information and waste computational resources. The middle section of the figure, where the model goes astray, is also interesting. After this split occurs, the observation that restores the $++++$ path to its "true" value is information-rich. However, the simulation has incurred a significant error by the time this observation is assimilated. If we could detect the divergence point indicated by the circle in the figure and apply the correction there, we could greatly improve the accuracy of the simulation.



(a) (b)

Figure 5: Assimilating data into the point-vortex model: The numerical results of Figure 4(b) are used to correct the vortices in the simulation of Figure 4(c). The solid line and the $+++++$ path are the true and corrected trajectories, respectively; the data-assimilation scheme corrects the latter to the former at the points indicated by the black squares. (a) displays the results when no correction is applied to the $+++++$ path and (b) displays the results of periodically correcting the "model" simulation at 25s intervals. The mean-squared error was 61.7 in (a) and 1.12 in (b).

These observations led us to develop a new scheme for timing vortex corrections, termed *dynamics-informed assimilation*, that attempts to identify dynamically sensitive regions. The goal is to correct the model *only* when the system dynamics indicate that a correction will be useful. When the model is highly accurate, the information content in the observations is fairly low—i.e., the assimilated observations do not impart a significant change to our prior knowledge of the system. In contrast, when the model is failing to track the true dynamics, the observations can drastically improve the simulation. If we can detect when the model might be diverging from reality, then we can intelligently select when to correct it. Though this appears obvious, it is a difficult task, as we do not know the "true" state of the system in practical data assimilation applications. Fortunately, we do know that solvers typically make mistakes in regions where velocity gradients are large. Tracking these gradients, then, provides information about the probability of model divergence at a given time in the simulation and is thus a useful indicator of correction importance. By correcting the model only when the gradients are large, we can target regions where correction is most beneficial, saving the computational cost of gathering and processing observations when they are not required.

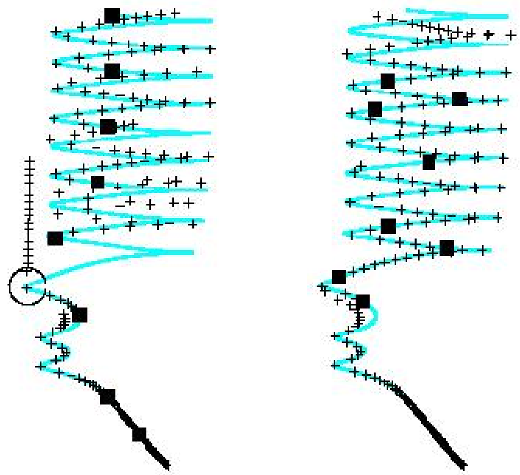
Our approach is as follows. At each timestep, we compute the components of the Jacobian of the velocity field at the location of each vortex using divided differences—that is,

$$J = \begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{pmatrix}$$

where u and v are the velocities in the x and y directions, respectively. The L_1 norm is then used to measure the size of these gradients.

The idea behind dynamics-informed data assimilation is to correct the vortices only when $||J||$ is high. To put this into practice, we had to develop a method to determine what values should be considered "high." Our first approach was to run the simulation twice: on the first run, we recorded the range of $||J||$ for each vortex. We then ran the experiment again, correcting the model when $||J||$ was in the top, say 20%, of its range for any given vortex. This method worked fairly well, but required a precomputation of $||J||$. This is not viable in real simulations, so we developed an on-line method that tracks $||J||$ and corrects the model when its increase from one timestep to the next exceeds a certain threshold percentage that we call J^+. Note that the value chosen for J^+ determines how many corrections are applied to the simulation. A larger value of J^+ neccessitates a larger increase in the norm of the Jacobian, which occurs less often. We can thus compare the performance of our dynamics-informed method to that of periodic correction by evaluating the success of each method when the same number of corrections is performed.

One such comparison is shown in Figure 6. In part (a) of the figure, we reproduce the periodic correction results from Figure 5(a). Figure 6(b) shows the results of dynamics-informed correction using the same datasets. In this figure, we can clearly see that our approach is working as desired. In the slowly varying region toward the beginning of the simulation, the model is doing quite well and so no corrections are applied. Looking at the circled area where the periodic case incurs the biggest error is also encouraging. The velocity gradients are quite large at this point, where the vortex is changing directions. Dynamics-informed correction captures this information and applies a timely correction to keep the vortex from going off-track. It may initially be confusing to observe that there are also some regions where it appears that the model is doing quite well, but a correction is still applied. In these low-gradient areas where the trajectory is fa! irly smooth, one would expect our approach to



(a) (b)

Figure 6: *Dynamics-informed* assimilation: These images show the same vortex (a) corrected periodically at 25s intervals and (b) corrected *only* when the norm of the Jacobian increases by 92 percent between timesteps. Notice that the second correction in (b) is applied at precisely the location where (a) goes "off track", resulting in a much more accurate simulation. The mean-squared error was 1.12 in (a) and 0.0491 in (b).

forego the assimilation. However, our approach is to correct *all* vortices whenever the velocity gradients at *any* vortex are increasing. Corrections in regions where the trajectory appears smooth are often due to high gradients in the vicinity of a different vortex.

Note that the simulations in Figures 6(a) and (b) have the same computational cost, in terms of the number of corrections applied. Recall that we can tailor the number of corrections performed by the dynamics-informed approach by tweaking the threshold percentage J^+ in the algorithm (the percentage increase in the norm of the Jacobian that warrants correction). Choosing a larger value for J^+ results in fewer corrections, while a smaller value results in more-frequent correction. To produce the results shown in Figure 6(b), we have chosen a J^+ value that results in the same number of corrections as in the periodic case in (a). This allows us to compare the accuracy achieved by the two simulations for the same computational cost. The MSE for the periodic correction approach was 1.12, while the MSE for the dynamics-informed simulation was 0.0491. This is a 23-fold improvement in accuracy for the same number of corrections!

These results may not be entirely convincing, since they involve a single vortex from a particular simulation. We have performed an ensemble of experiments using the symmetric and von Karman data sets, and the conclusion is the same: for the same number of corrections, dynamics-informed data assimilation results in much more accurate simulations than periodic correction. Figure 7 displays the results for the von Karman vortex configuration; here, we are plotting the log of the mean-squared error for each experiment based on the number of corrections applied[1]. We investigated periodic correction intervals of $50s, 100s, 150s, \ldots, 5450s$. Note that choosing a $50s$ correction interval performs a correction every time step, which will result in 0 MSE, while correcting every $5450s$ is the same as not correcting the simulation at all. The top curve in Figure 7 displays the results of these periodic correction experiments. To create the bottom curve, we ran simulations with Jacobian threshold percentages ranging from 0.5% to 300%; the resulting number of corrections in these simulations ranged from 1 to 13.

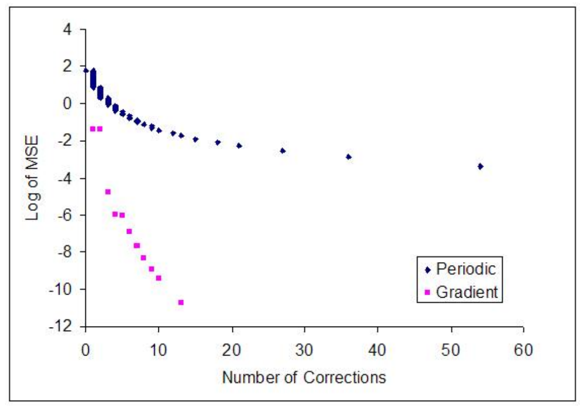


Figure 7: Comparison of dynamics-informed and periodic assimilation using the initial conditions in Figure 3(a). Each point in this figure represents a single simulation; the MSE is plotted as a function of the number of corrections. The upper curve displays the MSE results for an ensemble of periodic correction experiments; the lower curve displays the results when using the dynamics-informed correction strategy proposed here. The lower MSE values achieved by the latter indicate higher simulation accuracy.

This figure brings out several interesting features of the data assimilation process and provides some useful information about the von Karman vortex configuration in particular. For both periodic and dynamics-informed correction, the MSE decreases as the number of corrections increases. This matches our intuition about data assimilation, especially in this context in which the observations are perfect (i.e., noise-free): more corrections should generally produce a more-accurate result. When the error in the observations is significant, however—a common situation in the laboratory or the field—we may find that correcting more frequently is not always better. We are currently exploring this hypothesis in numerical experiments with noisy observations.

Other useful information can also be gleaned from the dynamics-informed curve in Figure 7. Note that the maximum number of corrections applied is 13, which occurs when the Jacobian threshold percentage is 0.5%. This means that the Jacobian of the velocity gradients increased by at least 0.5% for only 13 of the 108 time steps in this simulation. We can conclude that the velocity gradients in the

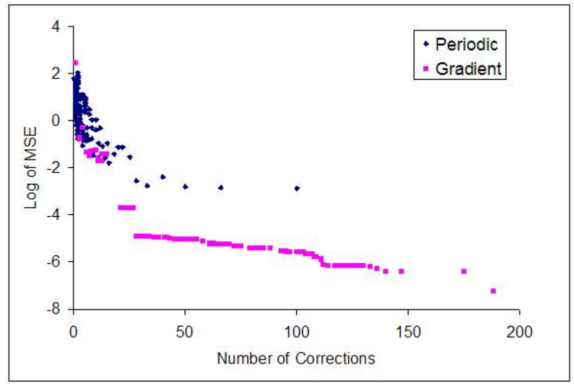
[1]Each point in the figure corresponds to a single simulation

von Karman experiment are very slowly varying. This is not entirely surprising, since the von Karman initial conditions were based on the stability criteria for a von Karman vortex street (with any instability resulting from the finite length of the street in our experiments). Also, the surprisingly low[2] MSE of 10^{-11} that results when these 13 corrections are applied strategically supports our contention that dynamics-informed assimilation works very well. In general, over all of the experiments, the differences in errors between the dynamics-informed and periodic simulations is quite dramatic. This encouraging result gives us confidence that this technique can also be applied successfully in real simulations with experimental data.

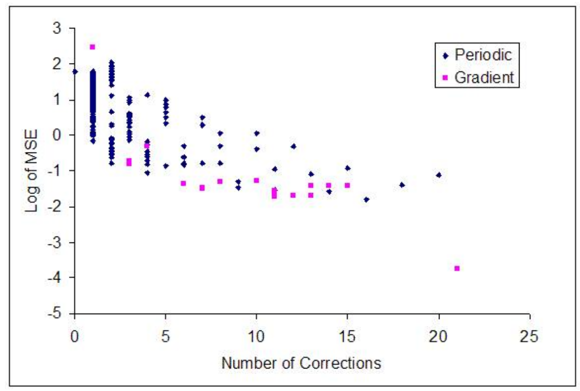
It is interesting to compare the von Karman results to those achieved with the symmetric vortex data sets from Figure 4. The MSE for each of the periodic correction experiments is again plotted in the upper curve in Figure 8(a). Recall that the correction intervals for these experiments ranged from $1s$ to $201s$, with the $1s$ correction interval resulting in 0 MSE and the $201s$ correction interval (uncorrected simulation) generating the largest error. For dynamics-informed correction, the threshold percentage was varied from 0.5% to 200%; these results are plotted in the lower curve in the figure. Here again, we see that correcting more frequently results in a more accurate simulation. However, the decrease in the MSE for dynamics-informed assimilation does not occur as rapidly as in the von Karman simulations. In spite of this, dynamics-informed assimilation outperforms periodic assimilation by a factor of 10 to 100 in most of the experiments.

The chart in Figure 8(b) also reveals some further interesting differences between the symmetric and von Karman experiments. Note that there are often several different MSE values that can result for a small number of corrections. For example, for a 200s simulation, there are ten different periodic correction intervals that result in 4 corrections—namely, $41s, 42s, \ldots, 50s$. Although each of these experiments performs the same number of corrections, the resulting MSE values are quite different, ranging from roughly 10^{-1} to 10^{1}. We also saw some similar variability in the von Karman experiments, but the MSE values were much more tightly clustered for a given number of corrections. This shows that the arbitrary choice of a periodic correction interval can have a serious impact on the simulation accuracy, especially in dynamically sensitive flows. Note that the dynamics-informed algorithm also requires a choice that affects the number o! f corrections performed: the value of Jacobian threshold percentage, J^{+}. One possibility is to dynamically modify J^{+} during the course of the simulation using an adaptive technique. If the model is significantly off-track when a correction is applied, it is likely that the threshold percentage is too high. Conversely, if corrections to the model are small, we can save computational resources by correcting less frequently (i.e., increasing J^{+}). Keeping track of the behavior of the norm of J over each correction interval will allow us to determine how much to increase or decrease our threshold. We are in the process of investigating this.

[2]The smallest double-precision value representable on the machine on which these experiments were performed was about 10^{-19}, so the MSE values in the figures are within the range of precision



(a)



(b)

Figure 8: Comparison of dynamics-informed and periodic assimilation using the initial conditions in Figure 3(b). Each point in this figure represents a single simulation; the MSE is plotted as a function of the number of corrections. The upper curve displays the MSE results for an ensemble of periodic correction experiments; the lower curve displays the results when using dynamics-informed correction. Part (b) of the figure zooms in on the leftmost region of (a), so results for smaller numbers of corrections can be seen more clearly.

Conclusion

We have proposed the use of a data-adaptive point-vortex model to overcome the speed and complexity limitations of current direct approaches to numerical simulation of complex fluid flows. In representing the flow only in terms of its coherent structures, the point-vortex model ignores all other dynamics, making it very fast. However, the point-vortex model is not nearly as accurate as DNS methods and is thus of questionable value in the context of real-time modeling and control applications. If this fault could be overcome, the point-vortex model could become a very powerful tool. Our solution to this problem is to correct that model with observations of the flow, a process known as data assimilation. The data assimilation algorithm must be developed with care, as an ineffective or computationally expensive approach would destroy the speed advantages of the point-vortex model.

We have presented a new correction methodology, which we call dynamics-informed data assimilation, that integrates quantitative information—sensor data—into this qualitative model. In our method, the correction timing is dictated by the underlying system dynamics: data is assimilated into the model only when the dynamics indicate that it is needed. In contrast to the standard periodic correction approach, our strategy targets dynamically-sensitive regions and avoids corrections when the model is performing well. Results from our initial experiments on this approach are quite encouraging. There is a significant increase in the accuracy of the simulation over standard periodic correction techniques. For the same number of corrections, there was typically at least an order of magnitude decrease in the mean-squared error. Stated differently, the dynamics-informed approach requires far fewer corrections to acheive the same simulation accuracy as periodic correction. This! novel result could allow the computational cost of gathering and processing system observations to be drastically reduced.

All of the experiments presented in this paper are numerical simulations with perfect observations. The ultimate goal of our research is to apply dynamics-informed data assimilation to real fluid flows. Clearly, measurements of any physical system will be contaminated with noise, which presents many additional challenges for any data assimilation strategy. These issues will be explored in our future work, and we will refine our dynamics-informed approach to ensure its utility for practical applications.

References

Airapetov, A. 1990. Motion of an initially point vortex in a flow of viscous fluid. *Journal of Applied Mathematics and Mechanics* 54:355–359.

Anderson, J. L. 2003. A local least squares framework for ensemble filtering. *Monthly Weather Review* 131(4):634–642.

Bailey-Kellog, C., and Zhao, F. 1997. Spatial aggregation: Modelling and controlling physical fields. In *11th International Workshop on Qualitative Reasoning*, 13–21.

Bailey-Kellog, C.; Zhao, F.; and Yip, K. 1996. Spatial aggregation: Language and applications. In *10th International Workshop on Qualitative Reasoning*, 3–11.

Basu, A.; Narasimha, R.; and Prabhu, A. 1995. Modelling plane mixing layers using vortex points and sheets. *Applied Mathematical Modelling* 19:66–75.

Berkooz, G.; Holmes, P.; and Lumley, J. 1993. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.* 25:539–575.

Boyland, P.; Stremler, M.; and Aref, H. 2003. Topological fluid mechanics of point vortex motions. *Physica D* 175:69–95.

Canuto, C.; Hussaini, M.; Quarteroni, A.; and Zang, T. 1988. *Spectral methods in fluid dynamics*. Springer.

Chavanis, P. 2001. Kinetic theory of point vortices: Diffusion coefficient and systematic drift. *Phys. Rev. E* 64:263091–263092.

Clancy, D. J., and Kuipers, B. 1993. Behavior abstraction for tractable simulation. In *7th International Workshop on Qualitative Reasoning*, 57–64.

Cortelezzi, L.; Chen, Y.; and Chang, H. 1997. Nonlinear feedback control of the wake past a plate: From a low-order model to a higher-order model. *Physics of Fluids* 9:2009–2022.

Daley, R. 1991. *Atmospheric Data Analysis*. Cambridge University Press.

Davies, H., and Turner, R. 1977. Updating prediction models by dynamical relaxation: An examination of the technique. *Quart. J. Roy. Meteor. Soc.* 103:225–245.

Dimet, F.-X. L., and Talagrand, O. 1986. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus* 38A:97–110.

Evensen, G. 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research* 99(C5):10,143–10,162.

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

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

Funakoshi, M. 1995. Evolution of vorticity regions of Karman-vortex-street type. *Fluid Dynamics Research* 15:251–269.

Germano, M.; Piomelli, U.; Moin, P.; and Cabot, W. 1991. A dynamic subgrid scale eddy viscosity model. *Phys. Fluids A* 3(7):1760–1765.

Huber, G., and Alstrom, P. 1993. Universal decay of vortex density in two dimensions. *Physica A* 195:448–456.

Ide, K., and Ghil, M. 1997. Extended Kalman filtering for vortex systems. Part I: Methodology and point vortices. *Dynamics of Atmospheres and Oceans* 27:301–332.

Ide, K.; Kuznetsov, L.; and Jones, C. 2002. Lagrangian data assimilation for point vortex systems. *Journal of Turbulence* 3:1–7.

ky Ringo Ling, S., and Steinberg, L. 1993. Approximation operators in distributed modeling. In *7th International Workshop on Qualitative Reasoning*, 138–144.

Lamb, S. H. 1945. *Hydrodynamics*. Dover Publications.

Lesieur, M., and Metais, O. 1996. New trends in large-eddy simulations of turbulence. *Ann. Rev. Fluid Mech.* 28:45–82.

Lundell, M. 1994. Qualitative reasoning with spatially distributed parameters. In *8th International Workshop on Qualitative Reasoning*, 187–196.

Lundell, M. 1995. A qualitative model of gradient flow in a spatially distributed parameter. In *9th International Workshop on Qualitative Reasoning*, 121–126.

Lundell, M. 1996. A qualitative model of physical fields. In *10th International Workshop on Qualitative Reasoning*, 121–126.

Moin, P. 1997. Progress in large eddy simulation of turbulent flows. Technical Report 1997-0749, AIAA.

Nishida, T. 1993. Generating quasi-symbolic representation of three-dimensional flow. In *7th International Workshop on Qualitative Reasoning*, 173–182.

Ordóñez, I., and Zhao, F. 2000. STA: Spatio-temporal aggregation with applications to analysis of diffusion-reaction phenomena. In *14th International Workshop on Qualitative Reasoning*.

Peacock, T.; Hertzberg, J.; Lee, Y.-C.; and Bradley, E. 2004. Forcing a planar jet flow using MEMS. *Experiments in Fluids* 37:22–28.

Riccardi, G., and Piva, R. 2000. Interaction of an elliptical patch with a point vortex. *Fluid Dynamics Research* 27:269–289.

Roshko, A. 1976. Structure of turbulent shear flows: A new look. *AIAA Journal* 14(10):1349–1357.

Sachenbacher, M., and Struss, P. 2001. AQUA: A framework for automated qualitative abstraction. In *15th International Workshop on Qualitative Reasoning*.

Sethian, J. 1991. A brief overview of vortex methods. In *Vortex Methods and Vortex Motion*. SIAM Press.

Struss, P. 2002. Automated abstraction of numerical simulation models—theory and practical experience. In *16th International Workshop on Qualitative Reasoning*.

Tarantola, A. 1987. *Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation*. Elsevier.

Yan, Y. 2003. Qualitative model abstraction for diagnosis. In *17th International Workshop on Qualitative Reasoning*.

Yip, K. 1995. Reasoning about fluid motion I: Finding structures. In *9th International Workshop on Qualitative Reasoning*, 201–207.

Yip, K. 1997. Structural inferences from massive datasets. In *11th International Workshop on Qualitative Reasoning*, 215–220.

Zhao, F. 1994. Intelligent computing about complex dynamical systems. In *8th International Workshop on Qualitative Reasoning*, 318–324.