A Realtime Observatory for Laboratory Simulation of Planetary Circulation*

S. Ravela, J. Marshall, C. Hill, A. Wong, and S. Stransky

Earth, Atmospheric and Planetary Sciences
Massachusetts Institute of Technology
ravela@mit.edu

Abstract. We present a physical, laboratory-scale analog of large-scale atmospheric circulation and develop an observatory for it. By combining observations of a hydro-dynamically unstable flow with a 3D numerical fluid model, we obtain a real-time estimate of the state of the evolving fluid which is better than either model or observations alone. To the best of our knowledge this is the first such observatory for laboratory simulations of planetary flows that functions in real time. New algorithms in modeling, parameter and state estimation, and observation targeting can be rapidly validated, thus making weather and climate application accessible to computational scientists. Properties of the fluid that cannot be directly observed can be effectively studied by a constrained model, thus facilitating scientific inquiry.

1 Introduction

Predicting planetary circulation is fundamental for forecasting weather and for studies of climate change. Predictions are typically made using general circulation models (GCMs), which implement the discretized governing equations. It is well-known that the prediction problem is hard [4]. Models typically have erroneous parameters and parameterizations, uncertain initial and boundary conditions, and their numerical schemes are approximate. Thus not only will the error between physical truth and simulation evolve in a complex manner, but the PDF of the evolving model state's uncertainty is unlikely to retain the true state within it. A way forward is to constrain the model with observations of the physical system. This leads to a variety of inference problems such as estimating initial and boundary conditions and model parameters to compensate for model inadequacies and inherent limits to predictability. Constraining models with observations on a planetary scale is a logistical nightmare for most researchers. Bringing the real world into an appropriate laboratory testbed allows one to perform repeatable experiments, and so explore and accelerate acceptance of new methods.

A well-known analog of planetary fluid-flow is a thermally-driven unstable rotating flow [2,3]. In this experiment a rotating annulus with a cold center (core) and warm periphery (exterior) develops a circulation that is dynamically similar to the mid-latitude circulation in the atmosphere (see Figure 1). We have built an observatory for this laboratory experiment with the following components: Sensors to take measurements of

* This material is supported NSF CNS 0540248.

Y. Shi et al. (Eds.): ICCS 2007, Part I, LNCS 4487, pp. 1155–1162, 2007.

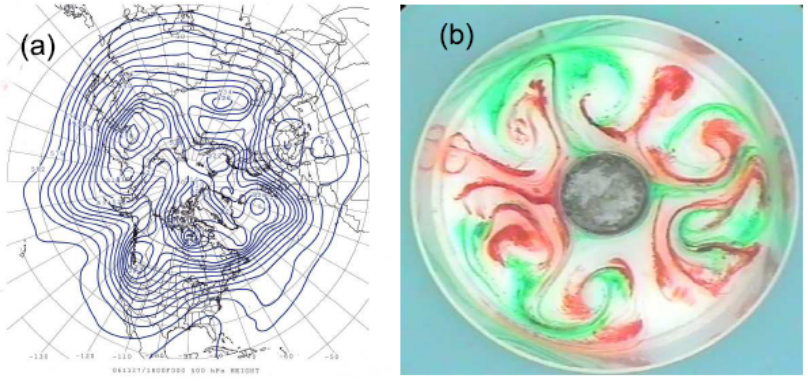

Fig. 1. Image (a) shows the 500hPa heights for 11/27/06:1800Z over the northern hemisphere centered at the north pole. Winds flow along the contours. Image (b) shows a tracer (dye) in a laboratory analog. The tank is spinning and the camera is in the rotating frame. Tracer droplets initially inserted at the periphery (red dye, warm region) and around the central chilled can (green dye, cold region) has evolved to form this pattern. The laboratory analog and the planetary system are dynamically akin to one-another. We study the state-estimation problem for planetary flows using the laboratory analog.

the evolving physical system, a numerical model trying to forecast the system, and an algorithm to combine model and observations. The challenges in building such a system are rather similar to the large-scale problem, in at least four ways. Nonlinearity: The laboratory analog is nonlinear and the numerical model is the same used in planetary simulations. Dimensionality: The size of the state of the numerical model is of the same order as planetary simulations. Uncertainty: The initial conditions are unknown, and the model is imperfect relative to the physical system. Realtime: Forecasts must be produced in better than realtime. This corresponds to a time of order ten seconds in our laboratory system within which a forecast-observe- estimate cycle must be completed.

In this report, we discuss the realtime system and focus on the problem of estimating initial conditions, or state. This estimation problem is posed as one of filtering and we demonstrate a two-stage assimilation scheme that allows realtime model-state estimates.

2 The Observatory

The observatory, illustrated in Figure 2, has a physical and computational component. The physical component consists of a perspex annulus of inner radius 8cm and outer radius of 23cm, filled with 15cm of water and situated rigidly on a rotating table. A robotic arm by its side moves a mirror up and down to position a horizontal sheet of laser light at any depth of the fluid. Neutrally buoyant fluorescent particles are embedded in water and respond to incident laser illumination. They appear as a plane of textured dots in the 12-bit quantized, $1K \times 1K$ images (see Figure 4) of an Imperx camera. These images are transported out of the rotating frame using a fiber-optic rotary joint (FORJ or slip-ring). The actual configuration of these elements is shown in a photograph of our rig in Figure 3.

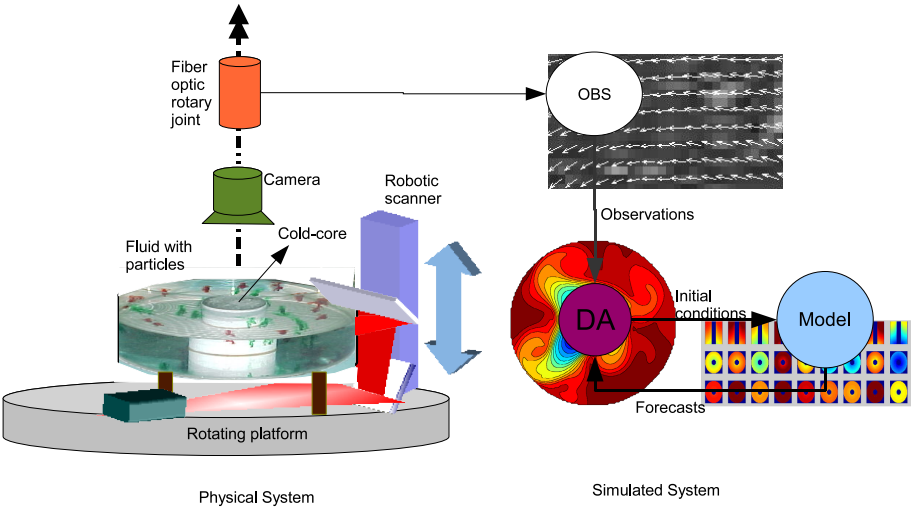

Fig. 2. The laboratory observatory consists of a physical system: a rotating table on which a tank, camera and control system for illumination are mounted. The computational part consists of a measurement system for velocimetry, a numerical model, and an assimilation system. Please see text for description.

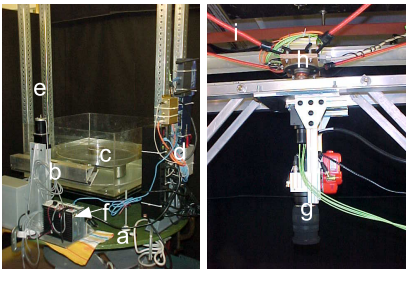

Fig. 3. The apparatus consists of (a) the rotating platform, (b) the motorized mirror, (c) the tank, (d) electronics, (e) a rig on which a camera is mounted, (g). Laser light comes from direction (f) and bounces off two mirrors before entering the tank. The fiber optic rotary joint (FORJ) (h) allows images to leave rotating frame and is held stably by bungee chords (i).

The computational aspects of the observatory are also shown in Figure 2. A server acquires particle images and ships them to two processors that compute optic-flow in parallel (Figure 2, labeled (OBS)). Flow vectors are passed to an assimilation program (Figure 2, labeled (DA)) that combines them with forecasts to estimate new states. These estimates become new initial conditions for the models. We now go on to discuss individual components of this system.

2.1 Physical Simulation and Visual Observation

We homogenize the fluid with neutrally buoyant particles and spin the rotating platform, typically with a period of six seconds. After twenty minutes or so the fluid entrains itself

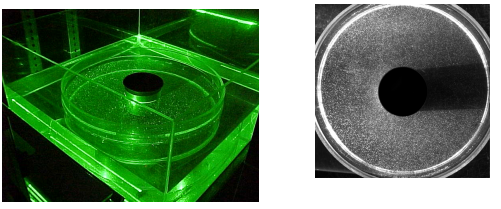

Fig. 4. The rotating annulus is illuminated by a laser light sheet shown on the left. The camera in the rotating frame sees embedded particles shown on the right. Notice the shadow due to the chiller in the middle. The square tank is used to prevent the laser from bending at the annulus interface.

to the rotation and enters into solid body rotation. The inner core is then cooled using a chiller. Within minutes the water near the core cools and becomes dense. It sinks to the bottom to be replenished by warm waters from the periphery of the annulus, thus setting up a circulation. At high enough rotation rates eddies form; flowlines bend forming all sorts of interesting structures much like the atmosphere; see Figure 1.

Once cooling commences, we turn off the lights and turn on the continuous wave 1W 532nm laser, which emits a horizontal sheet of light that doubles back through two periscoped mirrors to illuminate a sheet of the fluid volume (see Figure 4). An imaging system in the rotating frame observes the developing particle optic-flow using a camera looking down at the annulus.

The ultra-small pliolite particles move with the flow. We see the horizontal component and compute optical flow from image pairs acquired 125-250ms apart using LaVision's DaVis software. Flow is computed in 32×32 windows with a 16 pixel uniform pitch across the image. It takes one second to acquire and compute the flow of a single 1Kx 1K image pair. An example is shown in Figure 5.

Observations are gathered over several levels, repeatedly. The mirror moves to a preprogrammed level, the system captures images, flow is computed, and the mirror moves to the next preprogrammed level and so on, scanning the fluid volume in layers. We typically observe the fluid at five different layers and so observations of the whole fluid are available every 5 seconds and used to constrain the numerical model of the laboratory experiment.

2.2 Numerical Model

We use the MIT General Circulation Model developed by Marshall et al. [6,5] to numerically simulate the circulation. The MIT-GCM is freely available software and can be configured for a variety of simulations of ocean or atmosphere dynamics.

We use the MIT-GCM to solve the primitive equations for an incompressible Boussinesq fluid in hydrostatic balance. Density variations are assumed to arise from changes in temperature. The domain is three-dimensional and represented in cylindrical coordinates, as shown in Figure 6(a), the natural geometry for representing an annulus. In experiments shown here, the domain is divided into 23 bins in radius (1cm/bin), 120 bins in orientation (3^o bins). The vertical coordinate is discretized non uniformly using 15 levels and covering 15cm of physical fluid height, as shown in Figure 6(b). The fluid is modeled as having a free slip upper boundaries and a linear implicit free surface. The

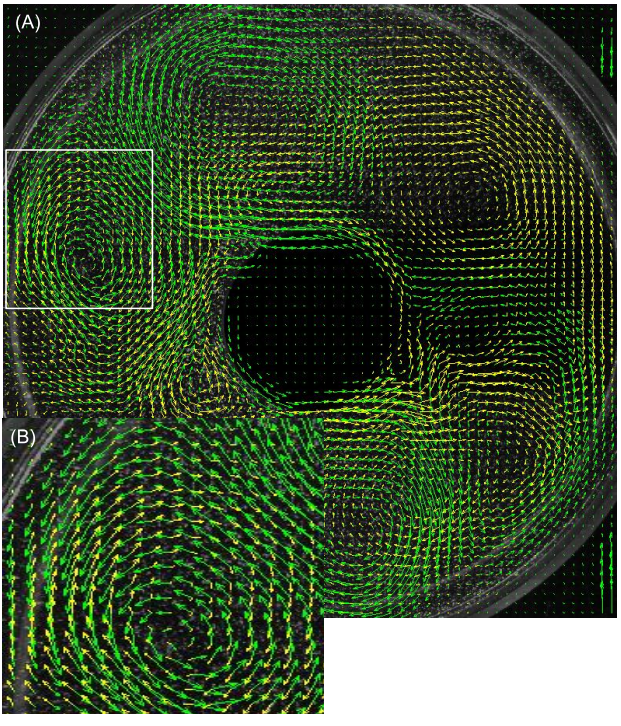

Fig. 5. A snapshot of our interface showing model velocity vectors (yellow), and observed velocities (green) at some depth. The model vectors macroscopically resemble the observations, though the details are different, since the model began from a different initial condition and has other errors. Observations are used to constrain model states, see section 2.1.

lateral and bottom boundaries are modeled as no-slip. The temperature at the outer core is constant and at the inner core is set to be decreasing with a profile interpolated from sparse measurements in a separate experiment (see Figure 6(b)). The bottom boundary has a no heat-flux condition. We launched the model from a random initial temperature field. A 2D slice is shown in Figure 6(c).

The MIT-GCM discretizes variables on an Arakawa C-grid [1]. Momentum is advected using a second-order Adams Bashforth technique. Temperature is advected using an upwind-biased direct space-time technique with a Sweby flux-limiter [7]. The treatment of vertical transport and diffusion is implicit. The 2D elliptic equation for the surface pressure is solved using conjugate gradients. In Figure 5, the model velocities are overlaid on the observed velocities after suitably registering the model geometry to the physical tank and interpolating. Despite the obvious uncertainty in initial conditions and other approximations, the model preserves the gross character of flow observed in the physical fluid, but at any instant the model-state differs from the observations, as expected.

The model performs in better than realtime. On an Altix350, using one processor, we barely make it[1], but on 4 processors we are faster by a factor of 1.5. The reason for this

[1] In ongoing work with Leiserson et al. we seek to speedup using multicore processors.

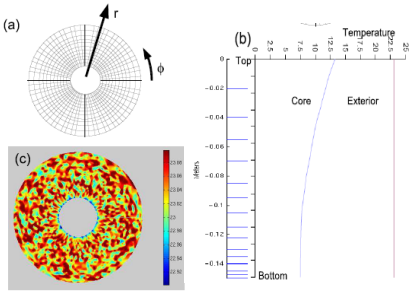

Fig. 6. (a) The computational domain is represented in cylindrical coordinates. (b) Depth is discretized with variable resolution, to resolve the bottom-boundary finely. The lateral boundary conditions were obtained by interpolating sparse temperature measurements taken in a separate run and the bottom boundary is no flux. (c) shows a random initial condition field for a layer.

performance is the non-uniform discretization of the domain using nonuniform vertical levels, which is also sufficient to resolve the flow.

2.3 State Estimation

An imperfect model with uncertain initial conditions can be constrained through a variety of inference problems. In this paper, we estimate initial conditions, or state estimation, which is useful in weather prediction applications. Following well-known methodology, when the distributions in question are assumed Gaussian, the estimate of state X_t at time t is the minimum of the following quadratic objective

$$J(X_t) = (X_t - X_t^f)^T B_t^{-1}(X_t - X_t^f) + (Y_t - h(X_t))^T R^{-1}(Y_t - h(X_t)) \quad (1)$$

Here, X_t^f is the forecast at time t, B is the forecast error-covariance, h is the observation operator and R is the observation error-covariance.

We use a two-stage approach. In the first stage, forecast and measured velocities at each of the five observed levels are separately assimilated to produce velocity estimates. Thermal wind is then used to map the implied temperature fields at these corresponding levels. Linear models estimated between model-variables in the vertical are finally used to estimate velocity and temperature at all model layers. Velocities at individual layers are assimilated using an isotropic background error covariance. In effect, we assume that the model does not have sufficient skill at startup and the first stage, therefore, is designed to nudge the model to develop a flow similar to the observations. Because the domain is decomposed into independent 2D assimilations and many independent 1D regressions, this stage is very fast.

After running the first step for a few iterations, the second stage is triggered. Here we use an ensemble of model-states to represent the forecast uncertainty and thus use the ensemble Kalman filter to compute model-state estimates. It is impractical to do large ensemble simulations. So we use time samples of the MIT-GCM's state and

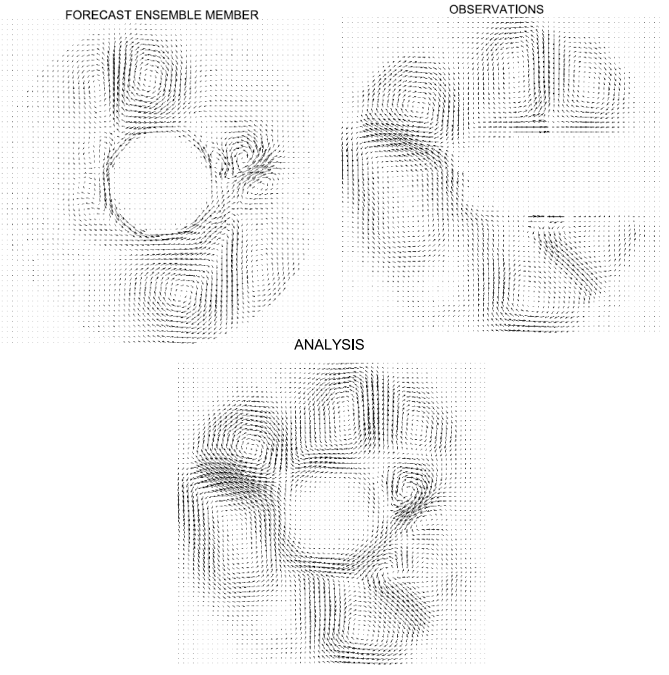

Fig. 7. The forecast velocity field at a time $t = 10min$ (left), observations at this layer (right) and estimated velocity (bottom). The shadow precludes measurements in a large area and produces noisy vectors at the shadow boundary.

perturb snapshots azimuthally with a mean perturbation of 0^o and standard deviation 9^o to statistically represent the variability of model forecasts. In this way the ensemble captures the dominant modes with very few numerical simulations. This method produces effective state estimates very efficiently.

The model is spun up from a random initial condition and forecasts 10 seconds ahead in approximately 6 seconds. Ensemble members are produced by time sampling every model second and perturbed to construct 40 member forecast ensembles. Planar velocity observations of 5 layers of the model are taken in parallel. In current implementation, assimilation is performed off-line though its performance is well within realtime. In under 2 seconds, the models and observations are fused to produce new estimates. The new estimated state becomes a new initial condition and the model launches new forecasts.

In Figure 7 the planar velocity of a forecast simulation, observations and estimates at the middle of the tank is shown 10 minutes into an experiment. As can be seen, the observations are noisy and incomplete due to the shadow (see Figure 4). The estimate is consistent with the observations and fills-in missing portions using the forecast. The error between the observations and estimates is substantially reduced. Please note that all 5 levels of observations are used and the entire state is estimated, though not depicted here for lack of space.

3 Conclusions

The laboratory analog of the mid-latitude circulation is a robust experiment, and the numerical model is freely available. Thus the analog serves as a new, easy-to-use, testbed. We have built a realtime observatory that to the best of our knowledge has not been reported before. Our hope is that the datasets generated here would find useful application to other researchers to apply their algorithms. Realtime performance is achieved through parallelism (observations), domain-reduction (model) and an efficient method to generate samples and compute updates (estimation).

A successful observatory also opens a number of exciting possibilities. Once the numerical model faithfully tracks the physical system, properties of the fluid that cannot easily be observed (surface height, pressure fields, vertical velocities etc.) can be studied using the model. Tracer transport can be studied using numerical surrogates. Macroscopic properties such as effective diffusivity can be studied via the model. For weather prediction, the relative merits of different state estimation algorithms, characterizations of model error, strategies for where to observe, etc etc, can all be studied and results reported on the laboratory system will be credible.

Of particular interest is the role of the laboratory analog for DDDAS. By building the infrastructure in the first year of this work, we can take on DDDAS aspects of this research in the second. In particular, we are interested in using the model-state uncertainty to optimize the number of observed sites, and locations where state updates are computed. In this way we expect to steer the observation process and use the observations to steer the estimation process.

Acknowledgment

Jean-Michel Campin and Ryan Abernathy's help in this work is gratefully acknowledged.

References

1. A. Arakawa and V. Lamb. Computational design of the basic dynamical processes of the ucla general circulation model. *Methods in Computational Physics, Academic Press*, 17:174–267, 1977.
2. R. Hide and P. J. Mason. Sloping convection in a rotating fluid. *Advanced Physics*, 24:47–100, 1975.
3. C. Lee. *Basic Instability and Transition to Chaos in a Rapidly Rotating Annulus on a Beta-Plane*. PhD thesis, University of California, Berkeley, 1993.
4. E. N. Lorenz. Deterministic nonperiodic flow. *J. Atmos. Sci.*, 20:130–141, 1963.
5. J. Marshall, A. Adcroft, C. Hill, L. Perelman, and C. Heisey. A finite-volume, incompressible navier stokes model for studies of the ocean on parallel computers. *J. Geophysical Res*, 102(C3):5753–5766, 1997.
6. J. Marshall, C. Hill, L. Perelman, and A. Adcroft. Hydrostatic, quasi-hydrostatic and nonhydrostatic ocean modeling. *Journal of Geophysical Research*, 102(C3):5733–5752, 1997.
7. P. K. Sweby. High resolution schemes using flux-limiters for hyperbolic conservation laws. *SIAM Journal of Numerical Analysis*, 21:995–1011, 1984.