

Kinematic Representation of Ocean Eddy *

Student: Dongwon ‘DW’ Han

Mentor: Dr. Elaine Spiller

August 4, 2012

Abstract

Ocean processes play an important role in local and global environments. Unfortunately, three dimensional ocean processes are poorly understood. Computational Fluid Dynamics, CFD, provides the most realistic simulations, but is computationally expensive. Kinematic Models, KM, are easy to evaluate, but cannot capture the same level of complexity.

In this project, we compare the Kinematic Model to the CFD. To do so, we use observations of the CFD fluid particle trajectories and assimilate those into the Kinematic Model to choose “good” parameter values. Even with a good choice of parameters, the kinematic model proves inadequate to capture the same level of complexity as the CFD. We characterize the difference between the CFD and KM velocity fields by modelling this difference as a random function. We see that a bias corrected KM model – the random function model plus the kinematic

*This work was supported by the National Science Foundation under grant #CNS-1063041.

model – captures a level of complexity far closer to the CFD than the Kinematic Model alone can achieve.

1 Introduction

Ocean plays an important role in local and global environments. Although there was significant effort to understand ocean processes, ocean processes are not yet understood. Because ocean processes are inherently complicated, scientists cannot solve problems involving these processes analytically. Furthermore, observation of ocean processes is difficult. Observations are often sparse in both time and space. Because these observations are time-averaged, it is hard to deduce precise dynamics from these observations. To overcome these difficulties, scientists use computer models to study them. Scientists can test their theories using these models. Also, computer models can handle data scarcity by interpolating observations.[11] In last few decades, two dimensional ocean models had many successes. These models gave deeper insight into physical and biological processes of ocean. However, two dimensional models have limited ability to describe three dimensional ocean processes. Hence, scientists and mathematicians aim to develop a fully three dimensional model. [5]

Computational Fluid Dynamics, CFD, is a system of partial differential equations solved numerically. CFD provides the most realistic representation of ocean processes. Unfortunately, CFD is computationally expensive. On the other hand, Kinematic Model, KM, is a system of three velocity functions of parameters x, y, z, α, β and ϵ_1 . KM is easy to compute. How-

ever, it does not capture the same level complexity as CFD. In this project, we compare KM to CFD. Also, we try to make KM to mimic CFD using statistical techniques, particle filter and random function.

2 Previous Work

KM is a system of three velocity equations. Each dimension has associated velocity function. Velocity functions are functions of six parameters: three Cartesian coordinates, x, y , and z , α, β and ϵ_1 .

$$U = -\frac{x}{3}(1 - 2z)(R - r) - \alpha y + \tilde{u}(x, y, z, \beta, \epsilon_1) \quad (1)$$

$$V = -\frac{y}{3}(1 - 2z)(R - r) - \alpha x + \tilde{v}(x, y, z, \beta, \epsilon_1) \quad (2)$$

$$W = z(1 - z)\left(\frac{2}{3}R - r\right) \quad (3)$$

U, V , and W are x velocity, y velocity and z velocity respectively. α controls the strength of horizontal velocity at the origin. α is twice the vertical component of the vorticity. ϵ_1 is a perturbation parameter. Finally, β controls how strongly vertical flow affects the horizontal velocity. [6, 7]

Two computer models are used to study ocean eddy. Ocean eddy is a circulating flow on scales of a few hundred kilometres. Eddies form when current is obstructed. Also, they form near the edge of permanent current.[11] Eddy is modelled as a rotating cylinder with radius one and height one. The can has a differential rotating lid which simulates cyclonic wind pressure on surface of ocean. At the top of the can, the rotating lid pushes fluid

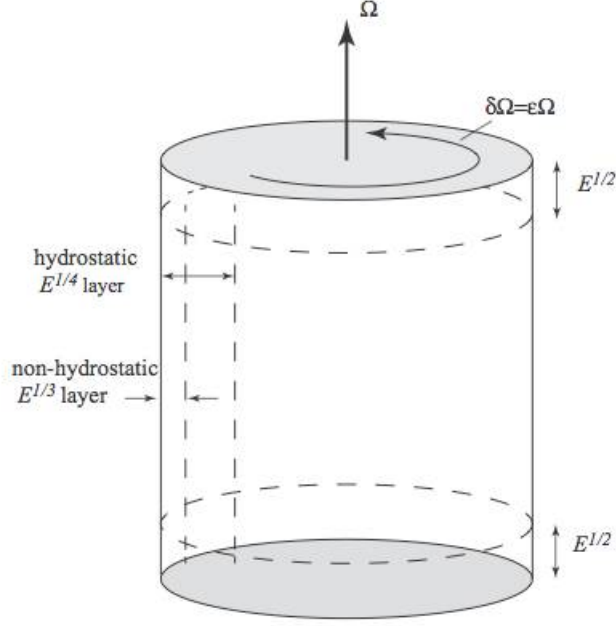


Figure 1: The rotating can [7]

outward toward sidewall. The sidewall redirects the flow to the bottom. Once the fluid reaches the bottom, it is recycled by a cyclonic circulation in the interior of the can; the circulation guides the flow to the top. All the velocities are zeros at the boundary of the can except at the lid. Total amount of the fluid inside of the can is conserved. Because the flow inside the can is linear and steady, all the fluid trajectories are periodic. In order to study mixing processes in three dimensions, this periodicity must be removed. Hence, velocity equations contain time varying perturbation terms, \tilde{u} and \tilde{v} . [6, 7, 8]

3 Methods

To compare KM to CFD, we use two statistical techniques; particle filter and random function. Statistical techniques have several advantages over deterministic techniques. Because model output can be estimated at untested output, statistical techniques can reduce the number of model runs. Also, these techniques naturally cope with data scarcity and provide a measure of accuracy of a computer model.[9]

Let y_{CFD} be the output of CFD and $y_{KM}(x, u)$ be the output of KM, where x is three dimensional Cartesian coordinates and u is other parameters of KM. We define error of KM as the norm of difference in velocity field between CFD and KM.

$$Error = y_{CFD}(x) - y_{KM}(x, u) \quad (4)$$

We want to minimize this error. One way to achieve this goal is to find optimum parameter values, \hat{u} , that minimize the error.

$$Error_1 = y_{CFD}(x) - y_{KM}(x, \hat{u}), \quad Error_1 \leq Error \quad (5)$$

To find optimum choice of parameters, we used particle filter method. Particle filter is a data assimilation technique. The goal of data assimilation is to use field/observation data to tune parameters of a computer model. Whenever observation is available, data assimilation updates probability distributions of parameters using Bayesian inference. Particle filter is also known as sequential Monte Carlo. Particle filter assimilates data that arrive

sequentially in time. Because observations often arrive sequentially in real life application, particle filter is an ideal technique for real time simulation. In this project, we treat CFD output as field data and assimilate the data into KM.[2][3][4][10]

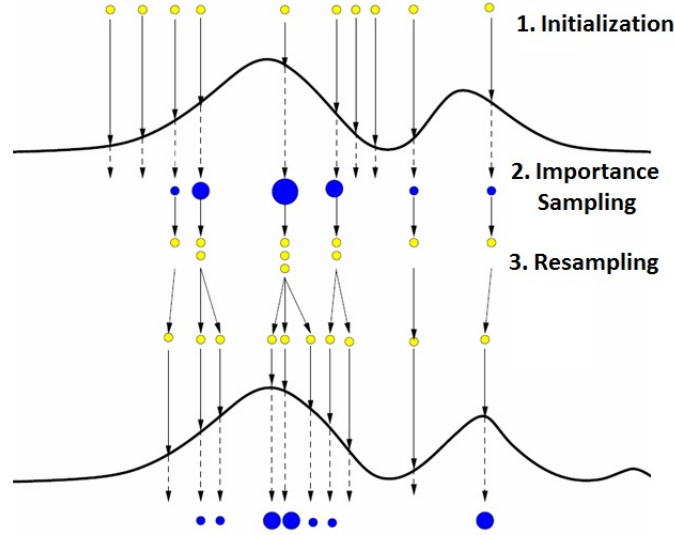


Figure 2: Pictorial representation of particle filter algorithm. Step 1. Initialization: A collection of N samples, or particles, are generated within observation uncertainty. Step 2. Importance Sampling: When new observation is available, each particle is assigned weight according to how close it is to the observed value. Step 3. Resampling: If weight is dominated by only few particles, generate a new collections of particles according to the importance weight. Step 4. Repeat 2-3 (Image source: [1])

To implement Particle Filter, we first generate observations from CFD output. At each observation time, we add noise to CFD data within observation uncertainty. Then we initialize a collection of 5000 samples, “particles”, within model uncertainty. Upon initialization, each particle has uniform weight. During the simulation, these particles move according to

KM. If new observation is available, particles get assigned weights based on how close they are to the observation value; this procedure is called importance sampling. After few importance samplings, samples usually degenerate: weights are dominated by few particles. When that's the case, particle filter perform resampling. In resampling, new collection of 5000 particles is generated. New particles are sampled from the previous collection proportionally to weights. Therefore particles with large weight are likely to be sampled multiple times. After multiple cycle of importance sampling and resampling, particles with "good" parameter values are expected to remian in the collection. [3, 4]

We also characterized error using random function. Given error at different inputs, random function approximates error at untested inputs, $b(x, u) \approx Error$. Random function naturally assimilate CFD data. And it work reasonably well with few inputs. Before collecting data, random function assume the distribution of error to be random. When we collect data, random function update the distribution using Bayesian inference[9]. Bias corrected KM is defined as the sum of random function and KM.

$$Error_2 = y_{CFD}(x) - (y_{KM}(x, u) + b(x, u)), \quad Error_2 \leq Error \quad (6)$$

Random function has its own set of parameters: mean, variance, theta, and p. We sample 150 points from CFD data. Random function then tune parameters for each bias function. After the tuning, properties of bias

functions match those of error. [9]

Finally, we analysed how sensitive KM output is to change in KM parameters. We generated “truth” run with following parameter values: $\alpha = 0.35, \beta = 1, \epsilon_1 = 0.45$. Then, all the parameters were fixed at “true” values, except one parameter that we changed in small increments. We quantified the relative error at grid points.

4 Results and Discussion

Typical particle filter run is shown in Figure 3. Sixty discrete points from CFD output were used to generate observations. However, we show continuous CFD trajectory, red line, in order to make comparison between two models easier. Red dot represent the location of the first observation. Green dot represent the location of the last observation. In every case we studied, KM trajectory usually do not match CFD trajectory. In general, mismatch is larger near the boundary than it is in the interior. We also observed that mismatch is large where velocity of CFD particle is high. Kinks on blue line is where particles are resampled. For a typical run, particles are resampled 20-30 times depending on initial observation of CFD data. Particle Filter re-sampled more often if the difference between KM velocity and CFD velocity is large.

Importance sampling assigns relative weights to the particle. Therefore, a particle that is not close to observation can get assigned large weight if it is closer to the observation than other particles. Because KM particles are often too far away from observations near the boundary, samples degenerates

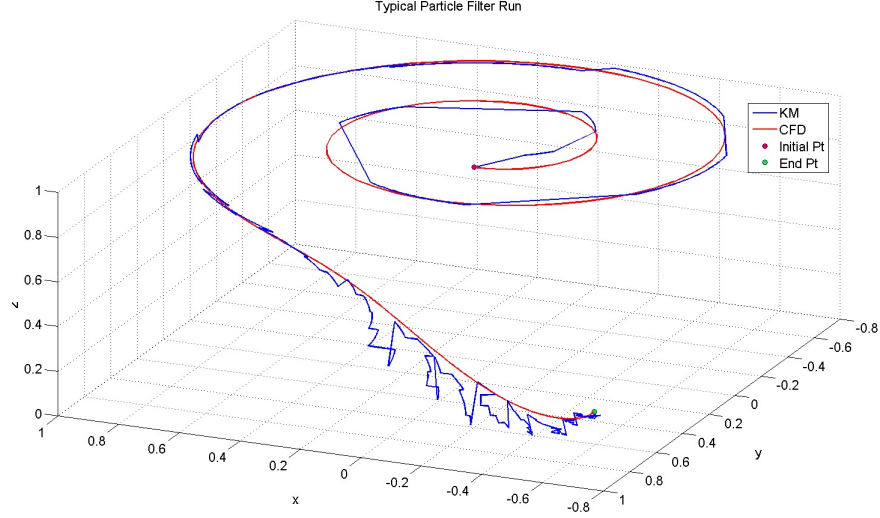


Figure 3: Three dimensional plot of typical particle filter run.

rapidly. Ideally, particle filter should find optimum choice of parameters. However, it failed to do in our case.

Figure 4 shows the distribution of alpha and beta values associated with particles after a complete cycle of particle filter. If particle filter is successful, we expect that alpha and beta values are concentrated at specific value. However, the distribution in Figure 4 are almost evenly spread out.

We also compared kinetic energy of KM to that of CFD. Figure 5 is kinetic energy comparison from typical particle filter run. Where CFD particle has low kinetic energy, the difference in kinetic energy is negligible. Because KM particles accelerate slower than CFD particle does, there are large difference in kinetic energy when CFD particle accelerates. Furthermore, because particles mostly accelerate near the boundary, the difference in kinetic energy is most notable near the boundary.

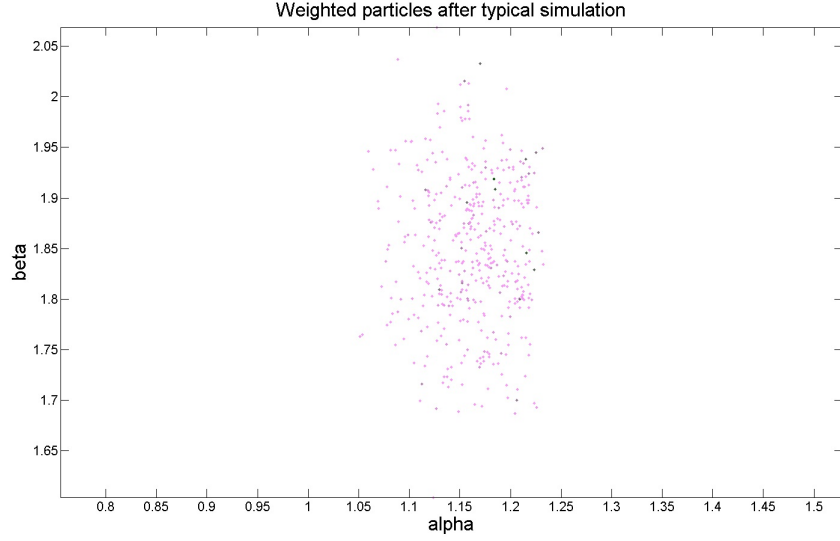


Figure 4: Distribution of alpha and beta values associated with particles after a complete cycle of particle filter

After a number of particle filter runs, we found that optimum parameter values do not converge. Depending on which CFD data we assimilated to KM, different optimum parameter values were estimated. Also, even with these optimum values, there was a substantial amount of error.

There are many possible sources of error. We postulated one of the major source of error is z velocity function, W . The mismatch in z velocity near the sidewall is large. Because z velocity function, W , is not a function of α, β and ϵ_1 , particle filter can not find optimum parameter values for W function. As consequence, KM model has large permanent error that can be reduced using particle filter.

The outcome of the sensitivity analysis is shown in 6. Relative error grows rapidly when α changes. Although change in ϵ_1 also introduces rela-

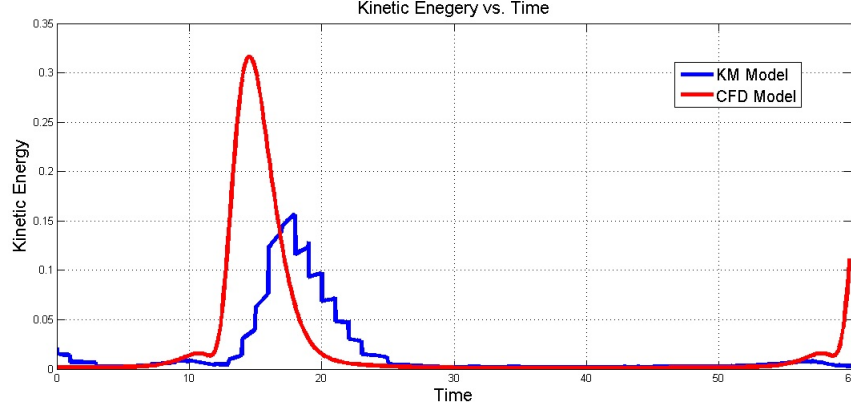


Figure 5: Comparison of kinetic energy

tive error, α dominates relative error at large deviation. These observations indicate that α is the most important parameter. Sensitivity analysis promoted us to ignore β and ϵ_1 when we generated bias functions using random function.

Figures 7 shows the velocity field of CFD and that of KM before bias correction. At the center region of the can, KM model does not capture the complex dynamics of CFD. Velocity of KM at the center region is too small compared to that of CFD. This small velocity make KM particle accelerates slower than CFD particle.

Figure 8 shows the images after bias correction. After the bias correction, KM field looks similar to CFD field. Unfortunately, bias-corrected field has unstable boundary. KM is streamlined to the boundary condition of the can. All velocities should reach zero at all boundary except at the lid. When bias functions are added to KM, they violate this zero velocity condition. To fix this problem, we sampled points at the boundary such that bias

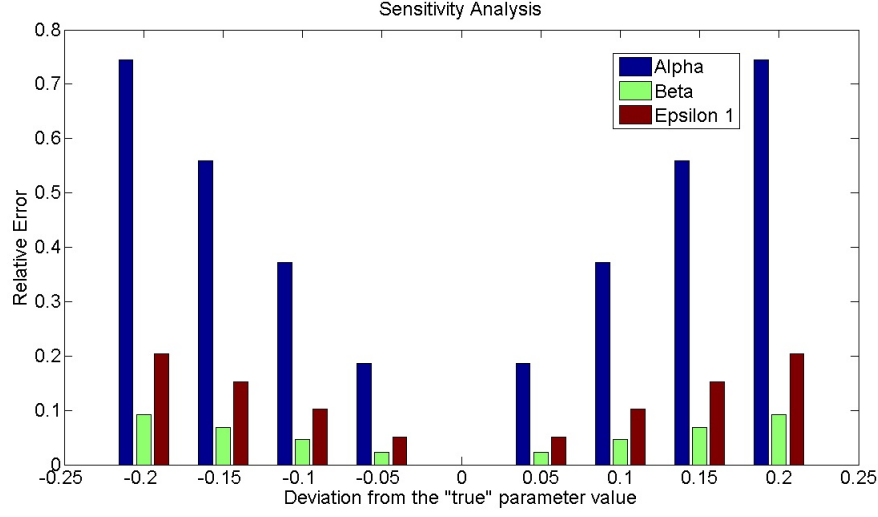


Figure 6: Truth run is generated with the following parameters; $\alpha = 0.35$, $\beta = 1$, and $\epsilon_1 = 0.45$. For each test, only one parameters are allowed to deviate from the true value; other parameters are fixed.

functions are tune to the boundary condition. Also, at the boundary, we uses KM without bias correction, while we use bias corrected KM in the interior. Between these two extremes, we take weighted average value of KM and bias corrected KM. These solutions are only temporary and we need a permanent solution for the boundary problem. After this problem is addressed, we want to generate poincare maps to check whether bias corrected KM indeed behaves like CFD.

References

- [1] Mobility and Handover Prediction Mechanism: a performance comparison exploiting several filters. Online, May 2010.
- [2] Arnaud Doucet and Adam M. Johansen. A Tutorial on Particle Filtering and Smoothing: Fifteen years later. Second Version, December 2008.
- [3] Arnaud Doucet, Nando de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*, chapter 1. An Introduction to Sequential Monte Carlo Methods. Springer, 2001.
- [4] Simon Godsill. Particle Filters. Lecture: Machine Learning Summer School, Cambridge, 2009.
- [5] Woods Hole Oceanographic Institution. Ocean 3d + 1. Online, April 2012. Online: accessed 20-June-2012.
- [6] Larry Pratt. Kinematic Representations of the Velocity Field.
- [7] Larry Pratt. The Rotating Can Problem. November 2011.
- [8] Larry Pratt, Tamay Ozgokmen, Yana Bebieve, and Irina Rypina. The Rotating Can Flow.
- [9] Jerome Sacks and William J. Welch. Design and Analysis of Computer Experiments. Lecture: Summer School on Computer Models and Geophysical Risk Analysis, August 2010.

- [10] Elaine Spiller. Brief Introduction to Probability and Simulation. Lecture: Monsoon Summer School, July 2011.
- [11] Robert H. Stewart. *Introduction to Physical Oceanography*. Orange Grove Texts Plus, 2009.

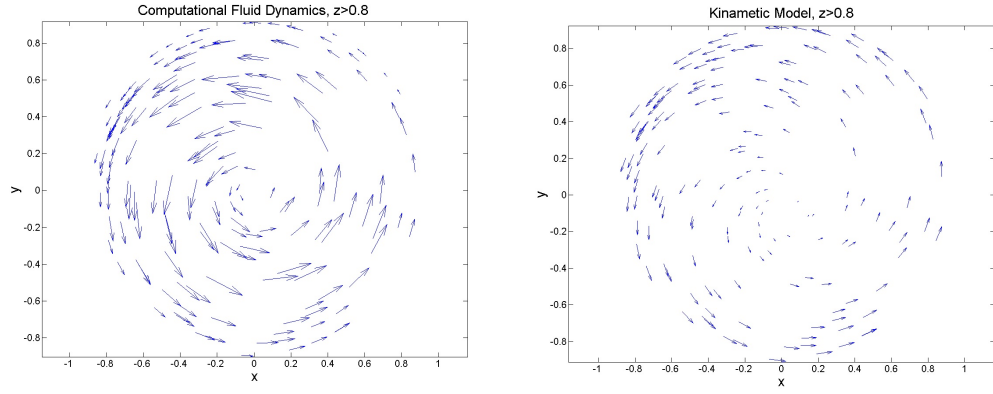


Figure 7: Left: CFD velocity field. Right: KM velocity field before bias correction

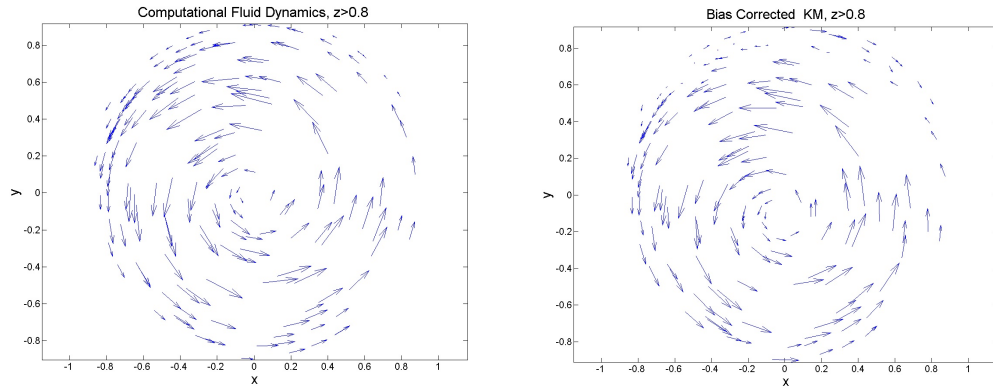


Figure 8: Left: CFD velocity field. Right: KM velocity field after bias correction