<u>Assignment 7 – When Bad Models Go Good</u>

Due: 19 December 2017 (via e-mail)

Objective

The objective of this assignment is to demonstrate the utility of data assimilation to correct for model forecast errors and thus provide an improved analysis for a subsequent forecast. Instead of using a computationally expensive full data assimilation and numerical model implementation, which has a steep learning curve, we instead apply the ensemble adjustment Kalman filter to our 1-D advection model from Assignments 2 and 3. In lieu of having actual real-world observations of our model variable *h*, we conduct an *observation system simulation experiment*, wherein we extract observations with specified error variance characteristics from a "truth" model simulation and then assimilate those into a set of ensemble forecasts.

Overview

For ensemble numerical weather prediction, there are many potential sources of model error: errors in the formulation of the model equations (e.g., the use of finite difference approximations), errors in physical parameterizations, and errors in the initial conditions. These all interact with each other in both linear and non-linear ways, resulting in erroneous (to varying extents) forecasts. Here, we will let the primary source of model error be with the finite difference approximation used in our model. We will use the centered-in-time, 2nd-order centered-in-space differencing scheme, which we have demonstrated to be particularly afflicted by implicit numerical damping and numerical dispersion, particularly for small values of the Courant number and short wavelength features. We will extract "truth" observations from a simulation using the Runge-Kutta 3, 6th-order centered-in-space differencing scheme.

The intent of the assignment is to see whether we can 'correct' for the poor model's performance by assimilating 'good' observations. We will vary the extent to which the observations are 'good' (their assumed error characteristics), how frequently we assimilate observations, and assimilated observation density to document sensitivity to the observation network. For ensemble numerical weather prediction, we would have the added complication of the observed variable not being at a location or of a quantity that the model explicitly predicts, but we will not consider that complexity in this assignment.

The general procedure for this assignment is as follows:

- *Run the "truth" model forward in time and extract "observations" (with specified error characteristics) from its forecast at desired locations and times.*

  The "observations" we extract are not exactly the "truth" model's forecast values at those grid points and times, however. Instead, we extract "observations" by assuming that they are randomly drawn from a normal distribution with mean equal to the model's forecast values and variance equal to the specified observation error variance. For instance, if the "truth" model simulation said that the value of a variable h at grid point 4 and time 10 was

5 m, and we specified that the observation error variance is 4 m^2, then our "observation" would be a random sample from a normal distribution with mean of 5 m and standard deviation of 2 m. The resulting "observation" will have a value that is close to, but not exactly, 5 m. MATLAB's *normrnd* and numpy's *random.normal* functions can accomplish this with appropriate inputs for the distribution's mean and standard deviation.

- *Create an ensemble of model initial conditions.*

  Whenever one first initializes an ensemble, the individual ensemble members must be generated in some fashion. For both this application as well as modern ensemble numerical weather prediction, we can take a single initial condition and randomly perturb it N times to obtain N ensemble members. For instance, consider an initial Gaussian wave of the form:

$$h(x) = a \exp\left[\frac{-(x-50)^2}{2c^2}\right] + d$$

  Here, a is the amplitude, x is the grid point, c is the standard deviation (controlling the width of the wave), and d is the amplitude shift. We could draw unique values – for each ensemble member, applicable equally at all grid points; not unique values at each grid point and each ensemble member – of a, c, and d from normal distributions with means of (100 m, 3 m, 0 m) and standard deviations of (1 m, 0.5 m, and 3 m). We do not want large perturbations here or in real-world applications – just large enough to achieve sufficient diversity across ensemble members. In this exercise, we will use this very setup to generate a 50-member set of ensemble initial conditions.

- *Run each ensemble member forward, assimilating observations at the desired locations and time steps.*

  We begin assimilating observations into our imperfect after the first model advance; e.g., we advance the model to time 2 and assimilate observations, then advance to time 3 and assimilate observations, and so on. To do so, we implement the ensemble adjustment Kalman filter algorithm from Assignment 6. We do so in the context of updating **all** model grid points at that time step, not just those where the observations are valid (e.g., akin to question 3 of Assignment 6). Note that observations can be assimilated sequentially (one at a time) – in other words, if you have four observations to assimilate at a given time, you can assimilate the first, then the second, then the third, and then the fourth.

- *Evaluate the model forecasts with and without data assimilation.*

  Departures of the ensemble forecasts – one set with and one set without data assimilation – from the "truth" model solution give us a measure of assimilation system performance. Qualitatively, creating plots of each and assessing them for differences provides one means of data assimilation performance evaluation. Quantitatively, subtracting the "truth" from each ensemble member and averaging it over all members, grid points, and times gives you the average bias. If you square the departures, take the square root of the result, and then average over all members, points, and times, you get the average root mean squared error. In this assignment, we focus more on qualitative evaluation and theoretical interpretation.

Summary of Model and Assimilation Setup

Both models will use a similar setup to that in Assignments 2 and 3: a grid of 100 points, with Δx = 10 km and periodic lateral boundary conditions; a duration of 100,000 s; and an initial wave starting point at grid point 50. Use a time step of 100 s, such that each simulation will proceed for 1000 time steps (or 1001 times).

The "truth" model initial condition uses a = 100 m, $c = \sigma$ = 3 m, and d = 0 m. It also uses a constant advective velocity of 10 m s^{-1}, such that the Courant number is 0.1. It uses the RK3-in-time, 6^{th}-order centered-in-space differencing scheme. All observations from the "truth" model simulation are randomly drawn from normal distributions with a mean equal to that of the grid point's value of h and a standard deviation of 2 m.

The ensemble uses the centered-in-time, 2^{nd}-order centered-in-space differencing scheme. Fifty ensemble members should be initialized and advanced. Observations are assimilated with an observation error variance of 4 m^2.

Instead of an identical advective velocity across all ensemble members, we use a unique advective velocity for each ensemble member that is drawn randomly from a normal distribution with a mean of 10 m s^{-1} and a standard deviation of 1 m s^{-1}. Note that the advective velocity for an ensemble member is constant in time; it only varies between ensemble members. This creates some variety in the ensemble member forecasts. The ensemble initial conditions are a Gaussian wave with a, c, and d randomly drawn from normal distributions with means of (100 m, 3 m, 0 m) and standard deviations of (1 m, 0.5 m, and 3 m), respectively. This results in variety in the ensemble member initializations. In all, the process we simulate with this model is nowhere near as stochastic as is the fluid atmosphere, but has the advantage of simplicity with regards to interpreting its output.

The ensemble adjustment Kalman filter, as specified in Assignment 6, is used for data assimilation. Include with your completed assignment the code for your "truth" model simulation (which should also perform observation extraction) and for your imperfect model + data assimilation simulation.

Questions

1. (5 pts) Run the "truth" model simulation. Create a plot of the model solution at times 500, 750, and 1001. Use a different color for each curve and provide a legend.

2. (5 pts) Generate and run the imperfect-model-based ensemble without data assimilation. Create a plot of the model solution for ensemble members 1, 5, 37, and 50 at time 1001. Please also plot the ensemble-mean solution at time 1001. Use a different color for each curve and provide a legend. Describe the general characteristics of each solution and how they vary between members.

3. Extract observations from the “truth” simulation at all times at four grid points: points 12, 37, 62, and 87. Assimilate these into the imperfect model ensemble at every time step after the first; e.g., advance the model to time 2, then assimilate data and advance the model to time 3, repeating through the last time.

   a. (10 pts) Create a plot of the ensemble-mean solution at times 101, 301, 501, 751, and 1001. Use a different color for each curve and provide a legend. How does the ensemble mean solution compare to that of any single ensemble member in #2? How well does data assimilation mitigate for model imperfections in this case?
   b. (10 pts) Create a plot of the ensemble variance by grid point at times 1, 101, 301, 501, and 1001. Use a different color for each curve and provide a legend. How does ensemble variance change with time? Based upon this information, to what extent will observation assimilation be able to correct for forecast errors at early times vs. at later times? Why? Can you use this to infer why the solution in #3a has the form that it does at those later times?

4. Extract observations from the “truth” simulation at all times at twenty grid points: points 2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62, 67, 72, 77, 82, 87, 92, and 97. Assimilate these into the imperfect model ensemble as in #3.

   a. (10 pts) Create a plot of the ensemble-mean solution at times 101, 301, 501, 751, and 1001. Use a different color for each curve and provide a legend. How well does the assimilation of many densely packed observations mitigate for model errors?
   b. (10 pts) Create a plot of the ensemble variance by grid point at times 1, 2, 5, 10, and 1001. Use a different color for each curve and provide a legend. How does the rate at which ensemble variance decrease compare to that in #3a? Will observations be able to correct for large model errors at later times for this dense of a network?
   c. (10 pts) Compare your plot from part (a) to that from Question 1 of Assignment 3, wherein you used the centered-in-time, 2nd-order centered-in-space scheme with U = 10 m s^{-1}, C = 0.1, a = 100 m, $c = \sigma$ = 4 m, and d = 0 m. How similar are the two results? Given your plot from part (b) and the ensemble formulation – notably that for U – in this assignment, describe why the two results (part (a) and Assignment 3) might be similar despite assimilating observations drawn from a “truth” model?

5. *One way by which background error variance reduction can be mitigated is by inflation, or artificially expanding the variance of the model background estimates at each assimilation time before any observations are assimilated (i.e., one inflation per time, not per assimilated observation). A spatially and temporally constant inflation formulation takes the form:*

$$h_{x,i}^{\text{inf}} = \bar{h}_x + \sqrt{\lambda}\left(h_{x,i} - \bar{h}_x\right)$$

Here, $h_{x,i}^{inf}$ is the variance-inflated value of h at grid point x for ensemble member i, $h_{x,i}$ is the pre-inflation value of h at grid point x for ensemble member i, $\overline{h}_x$ is the pre-inflation ensemble-mean value of h at grid point x, and λ is the non-dimensional inflation value. For λ = 1, the variance-inflated value $h_{x,i}^{inf}$ is simply equal to the pre-inflation value $h_{x,i}$.

   a. *(10 pts) Using λ = 1.01 to inflate the background estimates prior to assimilation, assimilate the extracted observations from grid points 2, 7, 12, 17, 22, 27, 32, 37, 42, 47, 52, 57, 62, 67, 72, 77, 82, 87, 92, and 97. Create a plot of the ensemble-mean solution at times 101, 301, 501, 751, and 1001. Use a different color for each curve and provide a legend. Describe how the solution differs from that in #4a.*
   b. *(10 pts) Create a plot of the ensemble variance by grid point at times 1, 2, 5, 10, and 1001. Use a different color for each curve and provide a legend. How does the rate at which ensemble variance decrease compare to that in #4b? How might this result in the improvement in model performance seen in part (a) versus that in #4a?*

6. *Another way to mitigate background variance reduction is to not assimilate observations at every time step. Indeed, the most rapidly-updating ensemble data assimilation system – currently the RIKEN system in Japan, assimilating radar data every 30 s on grids of 100 m to 1 km – only assimilates data every five (100 m) to fifty (1 km) time steps. This allows for meaningful background spread to redevelop (given a rather stochastic fluid atmosphere) before the next assimilation cycle reduces it once again.*

   a. *(10 pts) Repeat the assimilation exercise from #5a, except assimilating observations only every five time steps starting at t = 5. Create a plot of the ensemble-mean solution at times 101, 301, 501, 751, and 1001. Use a different color for each curve and provide a legend. Describe how the solution differs from that in #5a.*
   b. *(10 pts) Repeat part (a), except for λ = 1.05. How well is ensemble data assimilation able to correct for model error when this larger value of inflation is used?*