PHY2506, Fall 2015

Problem Set 2

Assigned Nov 12th, Due Nov 20th, 2015

1. Linear advection equation model. You must obtain the following MATLAB scripts: {oi, gauss, gcorr, getpsi, sqrwv, obspat, upwind}.m. In this problem, we will run an optimal interpolation (OI) scheme for a forecast model that is basically a passive tracer advection in a 1D periodic domain. The forecast model is simply

$$\frac{\partial u}{\partial t} + U\frac{\partial u}{\partial x} = 0$$

The x domain is [-2,2] and is periodic. The initial condition is a rectangular wave of the form:

$$u(x,t=0) = \begin{cases} 1, & -1 \le x \le 0 \\ 0, & \text{otherwise} \end{cases}$$

Using an upwind finite difference scheme, we can write the solution as

$$u_j = Cu_{j-1} + (1-C)u_j$$

where $C = U\Delta t / \Delta x$ is the Courant number and u_j is the numerical solution for $u(x = j\Delta x)$. Δx and Δt are the gridspacing and time step, respectively.

(a) Simulation experiments. Let us first examine the forecast model. Run the model alone (no data assimilation) by typing oi(0,1,1,1). This provides a Courant number of 1 and integrations from $T_0 = 0$ to $T_{\text{final}} = 1$. Obtain a plot of the initial and final states. Repeat this exercise for different Courant numbers of 0.95, 0.9, and 0.25. What is happening to the solution as the Courant number decreases?

(b) Now we will run an OI. We simulate the truth by running the forecast model. To simulate the observations, we perturb the truth by the observation error variance:

$$\mathbf{z} = \mathbf{Hx} + \mathbf{v}$$

The observations network will be simple. Observations are available every kth gridpoint on either the left half of the domain, or the whole domain. In time, observations are available every n timesteps. The analysis is generated whenever there is data, every n timesteps. Otherwise, the analysis reverts to the background state. Once a new analysis is obtained, the model is integrated forward by another timestep. Thus, we have an intermittent assimilation scheme.

(i) Run the OI scheme by typing oi(0,1,0.95,0). The Courant number will now be fixed at 0.95 and $T_{\text{final}} = 1$ again. Three questions will be asked. Hitting "return" will give the default. First enter observation frequency of 5 (obs every 5 timesteps), an observation sparsity of 1 (obs at every gridpoint), and "return' for the obs error standard deviation. This will give the default value of 0.02. How does the analysis compare with the truth? Does the error estimate make sense?

(ii) Now let's make the problem a little harder. Again type oi(0,1,0.95,0), but provide an obs error of 0.2. Keep the obs frequency of 5 and the obs sparsity of 1. What happened to the analysis and the error estimate?

(iii) Now let's see what happens when there are data gaps. Type oi(0,1,0.95,0), but answer "return" to all questions. This gives an obs every timestep, over the left half of the domain, with a standard deviation of 0.02. How does the analysis fare? Now decrease the observation frequency by typing first 2 then 5 and keeping the obs pattern and error standard deviation the same as before. Now what happens to the solution? Why?

2. Let us repeat the 1D advection problem from Problem 1 using a 3Dvar scheme instead of an OI scheme. Obtain var3d.m for this MATLAB exercise.

   (a) Compare var3d.m and oi.m. What are the relevant differences?

   (b) Under what conditions are OI and 3Dvar equivalent, in theory? Are these conditions satisfied in this problem?

   (c) The solution of the MAP cost function is given by

$$\hat{\mathbf{x}} = \mathbf{P}_x(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{z} + \mathbf{P}^{-1}\mu)$$

   where

$$\mathbf{P}_x^{-1} = \mathbf{P}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}.$$

   Determine the analysis error covariance matrix for 3Dvar. Add this calculation to the code (1 statement). (Hint: How is $\mathbf{P}^{\mathrm{a}} = \mathbf{P}_{\mathrm{x}}$ related to J ?)

   (d) Run the code. Type var3d(0,1,0.95,0). The Courant number will be fixed at 0.95 and T_{final} = 1. As before, three questions will be asked. Hitting "return" will give the default. First enter observation frequency of 5 (obs every 5 timesteps), an observation sparsity of 1 (obs at every gridpoint), and "return' for the obs error standard deviation. This will give the default value of 0.02. Compare the speed of 3Dvar and OI. Note that a qualitative comparison if sufficient. A quantitative comparison is not possible because the 3Dvar code includes the OI solution, for comparison purposes. Compare the solution to the OI case. Now, uncomment the code that uses Newton's method and comment the call to *fminsearch*. Compare the speed and accuracy again to OI. Note that Newton's method converges in 1 iteration because our cost function is purely quadratic. Which is faster, Newton's method or *fminsearch*? Is Newton's method feasible for large-scale applications?

   (e) Now let's see what happens when there are data gaps. Type var3d(0.1.0.95.0), but answer "return" to all questions. This gives an observation every time step, over the left half of the domain with a standard deviation of 0.02. How does the analysis fare? Now decrease the observation frequency by typing first 2 then 5 and keeping the observation pattern and error standard deviation the same as before. Now what happens to the solution? Why?

Newton Method solution for a quadratic cost function

Consider a quadratic cost function:

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} - \mathbf{b}^{\mathrm{T}}\mathbf{x} + c$$

where A is symmetric, positive semi-definite. The gradient is

$$\nabla J(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b} \ .$$

The Hessian, or second derivative, is

$$J''(\mathrm{x}) = \mathbf{A}$$

Since we want to minimize $J(\mathbf{x})$, we want to solve for $\nabla J(\mathbf{x}) = 0$. Now, for an initial guess $\mathbf{x}_g$, we have that

$$\nabla J(\mathbf{x}_g) = \mathbf{A}\mathbf{x}_g + \mathbf{b}$$

while at the solution $\hat{\mathbf{x}}$, the gradient is zero, i.e.,

$$\nabla J(\hat{\mathbf{x}}) = 0 = \mathbf{A}\hat{\mathbf{x}} + \mathbf{b}$$

Subtracting these two equation yields

$$\nabla J(\mathbf{x}_g) = \mathbf{A}(\mathbf{x}_g - \hat{\mathbf{x}})$$

Solving for $\hat{\mathbf{x}}$ we obtain

$$\hat{\mathbf{x}} = \mathbf{x}_g - \mathbf{A}^{-1}\nabla J(\mathbf{x}_g)$$

or on substituting the Hessian for A:

$$\hat{\mathbf{x}} = \mathbf{x}_g - (J'')^{-1}\nabla J(\mathbf{x}_g) \ .$$