Notes on "toy" analysis procedures

NATO ASI on Data Assimilation

R. Swinbank 10 May 02

These are a set of IDL/PV-wave procedures to show the results of analysing a small set of idealised observations, given a (one-dimensional) background field, and assumed (background and observation) error covariances. The aim is to allow you to explore the effect of varying the statistical parameters of simple data assimilation systems.

In each case, information from the observations ($\mathbf{y}_o$) is combined with information from a background field ($\mathbf{x}_b$) to produce an analysis field ($\mathbf{x}_a$) according to the standard OI (Optimal Interpolation) expression:

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{BH}^T\left(\mathbf{HBH}^T + \mathbf{R}\right)^{-1}\left(\mathbf{y}_o - \mathbf{Hx}_b\right).$$

B and **R** are the background and observation error covariances, respectively, and **H** is the observation operator (in these examples, just a simple interpolation). (Note that these IDL procedures explicitly invert the matrix, even though that is not usually the most efficient way of solving the equation.)

In each case, the procedure is invoked by the IDL command ".run *procedure_name*". The procedure will ask you various questions, such as where you want to put the observations, and the error characteristics of the observations and background field. The elements of the background error covariance **B** are given by

$$\mathbf{B}_{ij} = \sigma^2\mu_{ij},$$

where σ is the magnitude of background error. μ_{ij} is a second-order autoregressive correlation function of the distance r_{ij} between grid-points i and j (by default):

$$\mu_{ij} = \left(1 + r_{ij}/L\right)\exp\left(-r_{ij}/L\right),$$

where L is the correlation length scale. A similar expression is used for the elements of **R**, where the observation error correlations (if any) are assumed to depend on the distance between observations.

To retain the previous parameter settings just press ENTER; to alter a parameter enter the new value at the prompt. (Note that if you want to change an array, you will need to enter the correct number of new values.) The procedure will then plot a graph showing the observations, background and analysis. You then have the option of writing a postscript file (for a hard copy of the plot). Finally, you are given the option of repeating the calculation.

Analysis_2obs

This calculates an analysis from two observations. You can explore what happens if you change the observation values and locations. The graph shows what the analysis would be for each observation on its own, as well as for both together. What happens if you vary the background error correlation length L_f, (e.g. from 0. to 10.)? The effect of the observation error covariances is less obvious; what happens if the observation errors are highly correlated and the observations agree with one another? What happens if they disagree with one another? Try to explain what you find.

Analysis_mobs

This calculates an analysis of two sets of (up to) seven observations of each type. You can use one set to represent sparse, accurate, observations and the other set to represent dense, but inaccurate observations. Also, explore trade-offs between observation error magnitudes and observation error correlations.

Analysis_sim

This procedure simulates observations and the background field with specified error characteristics. You have several choices as to the "truth" field. The background is then generated from the truth field, either by adding on random perturbations (according to the background error parameters), or by shifting the truth field along the x-axis. The observations are generated from the truth field, again using the specified observation error statistics. One interesting exercise is to use the "step function" form of the truth field, with the background field equal to the truth, but shifted by a couple of units. Which parameter settings are best for getting the analysis to move the "step" back to the correct location?

The procedures as written are quite basic. Feel free to take copies and amend them to carry out your own explorations. In *analysis_sim*, you can easily amend the code to explore what happens if the assumed observation and background errors are different from their true values.