M.Sc. Course on Operational/Advanced Data Assimilation Techniques (MTMD02): Modelling a Homogeneous Background Error Covariance Matrix

R.N.B. January 2012

This computer practical is to demonstrate a common way of modelling a background error covariance efficiently for use in variational data assimilation. The matrix modelled here is univariate and homogeneous.

Univariate: this is the property of an error covariance matrix that treats errors in each variable separately (ie as uncorrelated). For example, if the errors in temperature and pressure are considered uncorrelated then the error covariance matrix is univariate in temperature and in pressure. The matrix considered in this practical is concerned with only one variable.

Homogeneous: this is where the error covariance between two points depends upon their separation only. In the Fig. below if the covariances are homogemeous then errors between the points A and B have the same covariance as between C and D. This is because each pair is separated by the same distance even though they are at different locations.

Figure 1: In a system with homogeneous error correlations, the covariance between A and B is the same as the covariance between C and D (A and B have the same separation as C and D).

In this practical you will assemble a piece of computer code (in the language Fortran-90) from subroutines provided to model the action of an error covariance matrix, but without knowing the matrix explicitly. Error covariance matrices play essential roles in operational data assimilation, but are often practically too large to store in the computer, so the ability to model a matrix is essential.

Theoretical results

The background term in the variational cost function is as follows

$$J_B[\mathbf{x}] = \frac{1}{2}[\mathbf{x}-\mathbf{x}_B]^\dagger \mathbf{B}^{-1}[\mathbf{x}-\mathbf{x}_B] \quad , \tag{1}$$

where **x** is the state vector, $\mathbf{x}_B$ is the background state, **B** is the background error covariance matrix and † means transpose and complex conjugate (a generalization of the transpose operation when dealing with complex numbers). Let us suppose that we can define a new vector χ that is related to **x** via the following transform (called a control variable transform)

$$\mathbf{x}-\mathbf{x}_B = \mathbf{U}\chi \quad . \tag{2}$$

Inserting (2) into (1) leads to the background term in the cost function in terms of χ

$$J_B[\chi] = \frac{1}{2}\chi^\dagger \mathbf{U}^\dagger \mathbf{B}^{-1}\mathbf{U}\chi \quad . \tag{3}$$

We have some freedom on our choice of **U**. Choose **U** such that

$$\mathbf{U}^\dagger \mathbf{B}^{-1}\mathbf{U} = \mathbf{I} \quad , \tag{4}$$

Which results in the background term of the cost function

$$J_{\mathrm{B}}[\chi] = \frac{1}{2}\chi^{\dagger}\chi \quad . \tag{5}$$

This choice means that elements of the new vector χ have background errors that are uncorrelated and have unit variance (i.e. the background error covariance matrix for χ is **I**). The **B**-matrix has disappeared from the cost function. In order to recover $\mathbf{x}-\mathbf{x}_{\mathrm{B}}$ however, **U** must be known – see (2). **U** may be derived from **B** via (4), but the whole exercise here is to avoid the need to know **B** explicitly. This is where the modelling of **B** comes into play – let us propose a **U** that has the following form

$$\mathbf{U} = \mathbf{F}^{\dagger}\mathbf{\Lambda}^{1/2} \quad , \tag{6}$$

where $\mathbf{F}^{\dagger}$ is the Fourier transform operator that transforms a spectral-space vector to a real-space vector (see "Mathematical Tools" handout) and $\mathbf{\Lambda}$ is a diagonal matrix of weights (the diagonal elements form a function of wavenumber called the variance spectrum). Form (6) assumes that waves of different wavenumber (or wavelength) in the system are uncorrelated.

In order to test form (6) we would like to know what the equivalent **B**-matrix is. Rearranging (4) leads to

$$\mathbf{B} = \mathbf{U}\mathbf{U}^{\dagger} \quad . \tag{7}$$

This is called the implied **B**-matrix. Thus minimizing a cost function with background term (5) with respect to χ (and then transforming the result to real-space with (2)) would give the same result as minimizing a cost function with background term (1) with respect to **x** where **B** is defined by (7). Minimizing the former is simpler as an explicit **B**-matrix is not needed – we need know only how to act with the operator **U**. It is known that the **B**-matrix implied when using form (6) for **U** has the homogeneous property (later in the lectures this will be shown to be the case analytically, although you can demonstrate this here numerically).

Your tasks

A set of Fortran-90 subroutines are provided to perform the operations **U** and $\mathbf{U}^{\dagger}$. You will choose the shape of the variance spectrum and look at the result of acting with $\mathbf{U}\mathbf{U}^{\dagger}$ on a chosen state. First set-up the code and do a test run.

1. Download the code `homoegeous.f90` (see below) and examine the variables and the subroutines.
2. Assemble the code – you will need to call subroutines called `SetUp`, `CVT_hat` and `CVT`. These subroutines set-up some arrays (including the variance spectrum), act with the adjoint of the control variable transform operator $\mathbf{U}^{\dagger}$ and the control variable transform operator **U** respectively. Make sure that you understand the inputs and outputs of these subroutines.
3. Choose a form of the variance spectrum inside subroutine `SetUp`. A suggestion is to use a Lorentzian form as a function of wavenumber as follows

$$\mathbf{\Lambda}(k) = \frac{\alpha}{1+(k/L)^{2}} \quad , \tag{8}$$

where *k* is wavenumber (the wavenumbers are stored inside array called `Wavenumbers`), *L* is the spectral width of the Lorentzian and α is chosen as to satisfy the following normalization

$$\frac{1}{N_x}\sum_{1}^{N_x}\mathbf{\Lambda}(k) = \sigma_{\mathrm{B}}^{2} \quad , \tag{9}$$

where there are N_x grid points (and wavenumbers) and σ_{B} is the background error standard deviation. The notation $\mathbf{\Lambda}(k)$ in (8) and (9) means the diagonal element of matrix $\mathbf{\Lambda}$ that corresponds to wavenumber *k*. Note that in the code the variance spectrum is actually stored in a one-dimensional array that represents the diagonal of Λ .)

4. The array called `InputState` is fed into the string of operators $\mathbf{U}\mathbf{U}^{\dagger}$ and the output state is called `OutputState`. Set a form of the input array (a good choice is to set it to zero everywhere except for unity at a chosen point somewhere in the domain).
5. The code outputs a number of files which contain (amongst other things) the output state. Plot the output state for the given input state and variance spectrum.

Example things to try:

- Try changing the input state (e.g. move the position where the unit point is) and see the effect on the output state. Is the implied **B**-matrix homogemeous?
- Try making the variance spectrum broad in spectral space and then narrow.

How to download and compile the Fortran-90 code

Go to the blackboard web page for this course and download the following file from the 'Assignments' section

```
homogeneous.f90
```

Alternatively, the code is available via the web address

```
www.met.rdg.ac.uk/~ross/MTMD02/homogeneous.f90
```

You are free to use the computer system of your choice. For Meteorology Dept. UNIX, the compilation command is given in the comments at the start of the code. To compile and run the code in the PC lab follow this procedure.

- Download the modules to an appropriate place (e.g. the N: drive on the PCs connects to your central university account) and assemble the code.
- Bring up the FORTRAN package (select `Start → All Programs → Programming → Salford Software → Salford Plato IDE`).
- Load `homogeneous.f90`.
- To compile, select `Project → Compile File`.
- To run, select `Project → Run`.