Four Dimensional Data Assimilation Practical Exercises

1 The 1963 Lorenz Model: Nonlinear Dynamics and Sensitivity to Initial Conditions

The 3-variable model of Lorenz [1963] is frequently used to test data assimilation algorithms because, like the real atmosphere, it displays deterministic chaos.

$$\frac{dx}{dt} = f_1(x,y) = \sigma(y-x) \quad (1)$$

$$\frac{dy}{dt} = f_2(x,y,z) = \rho x - y - xz \quad (2)$$

$$\frac{dz}{dt} = f_3(x,y,z) = xy - \beta z. \quad (3)$$

We will use this model in a set of data assimilation experiments, with the following parameters:

$\sigma = 10$ –the Prandtl number
$\rho = 28$ –the normalized Rayleigh number
$\beta = 8/3$ –a nondimensional wavenumber.

1.1 Running the Lorenz model in Matlab

The Matlab program `EnKF_l63.m` runs the Lorenz model forward given a set of initial conditions, along with a forecast ensemble. It also runs an optional ensemble Kalman filter (EnKF) that assimilates regularly-spaced observations of some or all the model variables, but first we'll focus on the model. The model and assimilation inputs are set in `set_enkf_inputs_template.m`. Copy this file to a new file called `set_enkf_inputs.m`, then open this file in an editor and set `run_filter` to zero. Now

```
> E = set_enkf_inputs
```

sets the input variables and parameters. The output, E, is a matlab structure that contains all the model and assimilation parameters needed to run the EnKF. Now to run the model, type

```
> A = EnKF_l63(E)
```

The code produces a few plots that show the three model variables in time, comparing the truth and a model ensemble. The output A is a matlab structure that holds the truth, observed, and analysis (this will come later) of the three model variables, as well as a few other things (look at the first few lines of `EnKF_l63.m` to see what they are).

Try runnning the model with a few different initial conditions (by re-running `set_enkf_inputs.m` or simply changing $E.xt0$) to get an idea of the overall behavior of the model. You can plot the famous "Lorenz" butterfly by entering

```
> plot3(A.xt,A.yt,A.zt)
```

You'll see that the Lorenz model, like the weather, is chaotic – small changes in the initial conditions become huge changes down the line.

You'll also see that the ensemble (gray) eventually separates so much from the truth that its mean (green) no longer looks like a typical state of the Lorenz model (i.e. it's no longer on the "strange attractor" of the model). If the Lorenz model was a model of the weather, taking the ensemble mean as a weather forecast would be a pretty bad idea – not only would it be far from the truth, it wouldn't even be physical. Fortunately, in the real world we have observations of the weather to correct our forecast models and bring them closer to the truth. This will be illustrated in the next section.

1.2 True versus estimated error

`EnKF_l63.m` doesn't just produce a model trajectory, but also a forecast ensemble. The mean of this ensemble may be more or less close to the truth, but what the ensemble also tells us about is how certain or uncertain our forecast is. But how well does the ensemble really represent what is going on? The output structure `A` contains two quantities that can help us answer this question:

`A.EAave` is the time-averaged "analysis error" of the ensemble, i.e. error that we estimate using the ensemble standard deviation.

`A.ETave` is the root-mean-square *true error*, that is, the difference between the truth and the ensemble mean.

In a good ensemble prediction system (and later, ensemble filter), the ensemble spread should be similar to the true error. In the exercises to follow, keep an eye-out for how well these two fit together.

2 The Ensemble Kalman Filter

2.1 How much information is needed to capture the right statistics?

Now turn on the EnKF assimilation by setting `run_filter=1` in `set_enkf_inputs.m`. Other settings for the filter are explained in the comments of `set_enkf_inputs.m`. You can also change them directly in the structure, like (for example)

```
> E.run_filter = 1
```

Here we will test different values of the observation interval (`tobs`) and ensemble size (`N`).

Exercise 2.1.1: Run the filter with various ensemble sizes and a constant observation interval of 3. How many ensemble members are need to capture the true state?

Exercise 2.1.2: Run the filter with various observation intervals and an ensemble size of 10. How long can we go between observations and still capture the truth?

2.2 Observed vs. Unobserved variables

Here we will investigate how the Ensemble Kalman Filter transfers information from observed to unobserved variables. To turn off which variables are observed, change `obsx`, `obsy`, and `obsz`. In all exercises below, unless stated otherwise, use an observation interval `tobs=1` and ensemble size `N=10`. Try many different initial conditions for each case.

Exercise 2.2.1: Run the filter with observations only of x, only y, and only z. Which observation(s) gives us the best results?

Exercise 2.2.2: Run the filter with observations of x and y together, but not z. Compare this to the case where only z is observed. How good is the analysis of z in each case?

2.3 Multivariate versus Univariate Assimilation

A major advantage of four-dimensional assimilation algorithms like the EnKF is that they use the assimilating model to estimate the correlations between different model variables. To see the difference that this makes, we can manually force the inter-variable correlations in the EnKF covariance matrix to zero by setting `localize = 1`.

Exercise 2.3.1: Compare the effect of localization for analyses of z when we observe only x and y, for `tobs = 1,2,3,4,5`

2.4 Non-identity Observations

In the real world, we often don't observe model variables themselves, but functions of the model variables. Let's look at what happens in the Lorenz model when we observe functions of the three variables. This can be done by setting

```
E.obsx = 0
E.obsy = 0
E.obsz = 0
E.obs_meanxy = 1
E.obs_meanyz = 1
E.obs_meanxz = 1
```

Exercise 2.4.1: Run the filter with observations of mean(x, y), mean(x,z), and mean(y,z) and compare to observing the variables normally.

Exercise 2.4.2: Try assimilating observations of mean(x, y), mean(x,z), and mean(y,z) for different observation intervals.

References

E. N. Lorenz. Deterministic Nonperiodic Flow. *J. Atmos. Sci.*, 20:130–141, 1963.