Outline – Day 2

EF5 OVERVIEW

- Features of EF5
- Model structure
- Control file options
- Warm-up and model states
- Model evaluation indices

DEM DERIVATIVES

RAINFALL AND PET

AUTOMATIC CALIBRATION
Features of EF5

EF5 is the Ensemble Framework for Flash Flood Forecasting

EF5 is a distributed hydrological model
It takes advantage of multiple computing cores and parallel computing strategies for faster simulations

EF5 supports ensemble forecasting
Multiple model cores exist in EF5 and more will be added in the future
Each core contains different physics
The user can thus see multiple model solutions for a particular scenario
These multiple solutions are possible even with only one set of input data – this saves time!
Features of EF5

**EF5 is cross platform**

This training is designed for Microsoft Windows
EF5 was originally developed on Unix-like systems
This includes Linux distributions and Mac OS X

**EF5 is user-friendly**

As you saw in the Wang Chu example, if all the required data is provided, running EF5 is as simple as a double-click
EF5 is flexible and can accept multiple input data formats
If something goes wrong, EF5 includes helpful error messages to track down the problem
EF5 includes coupling

Runoff generation and flow routing are coupled together which makes the simulation more realistic.
EF5 output can be used to study atmospheric water, surface water, and subsurface water.

EF5 is scalable

Water storage in the soil is modeled at the sub-grid scale, as are runoff generation processes, so you can think of the cells as “self-contained”, which means the number of cells and the size of the cells can be specified by the user.
Suitable for flash floods at small scales, riverine floods at very large scales, or global forecasting.
EF5 is broken down into two main sections

Water balance
Routing

The water balance concerns the water inputs and outputs for each model cell

Recall that the inputs (precipitation, upstream runoff, and interflow) must equal the outputs (runoff and interflow to be routed downstream)

Routing determines how quickly these outputs will travel downstream
Right now, EF5 has two options for the water balance with another under development:

- CREST – used in all training examples
- SAC-SMA (Sacramento-Soil Moisture Accounting)
- HyMod (in development)

There are two choices for routing:

- Kinematic wave – used in all training examples
- Linear reservoir (in development)

Other available model components:

- SNOW-17
- Simple inundation
The EF5 control file tells the model what to do. It consists of several blocks, each of which starts with a [bracketed statement]. Types of blocks include:

- Basic
- PrecipForcing
- PETForcing
- Gauge
- Basin
- etc…
In each bracketed statement, the type of block should not be changed.

But the user can change the name of each block:

- TRMM
- FEWSNET
- Chhukha
- Wangchu
- etc…
The “Basic” block contains six pieces of information

- DEM, DDM, and FAM
  The file path to these topographical grids
- PROJ
  This is the map projection associated with the topographical information
- ESRIDDM and SelfFAM
  These are options related to how topographical information is encoded

More on all this in Module 2.2
PrecipForcing Block

This “PrecipForcing” block is named “TRMM” in this example

- **TYPE**
  The file type of the precipitation
- **UNIT**
  The units of the precipitation
- **FREQ**
  The frequency at which EF5 will ingest precipitation files
- **LOC**
  The directory where the precipitation is located
- **NAME**
  How precipitation files are named (TR_20150101.bif, TR_20150102.bif, etc..)

More on all this in Module 2.3
This “PETForcing” block is named “FEWSNET” in this example

- **TYPE**
  The file type of the PET
- **UNIT**
  The units of the PET
- **FREQ**
  The frequency at which EF5 will ingest PET files
- **LOC**
  The directory where the PET is located
- **NAME**
  How PET files are named (PET025.01.bif, PET025.02.bif, etc.)

More on all this in Module 2.3
Multiple “Gauge” blocks can be specified, one for each gauge you want to run the model at – this one is named “Chhukha”

- **LON and LAT (or CELLX and CELLY)**
  The longitude and latitude of the gauge (unprojected and in decimal degrees)
  (or the x- and y- grid coordinates of the gauge in the topographical basic files)

- **OBS**
  Location of a file with observed stream flow values (required for calibration)

- **BASINAREA**
  Contributing area of the gauge in square kilometers

- **OUTPUTTS**
  Output a time series file for the gauge
Multiple “Basin” blocks can also be specified – this one is named “Wangchu”

- **GAUGE**

  The name of the gauge block to be included in this basin

Note that the term basin here is not a physical basin

Basin here means a collection of gauges (or gauge blocks) to be modeled in a single run of EF5
Each component of the model has its own set of parameters

Parameters are like the knobs on a car radio. They allow us to “tune” the model to get better simulations, like how you tune the radio to get better reception.

Let’s start with the CrestParamSet

6 parameters
The CrestParamSet block in this example is named “Wangchu”

- **GAUGE**
  The name of the gauge block to which these parameters apply

- **WM**
  Maximum soil water capacity

- **B**
  The exponent of the VIC

- **IM**
  Impervious area ratio

- **KE**
  Conversion factor from PET to actual ET

- **FC**
  Soil saturated hydraulic conductivity

- **IWU**
  Initial value of soil water, a % of WM
If you want to specify different parameters for different gauges, you can include additional CrestParamSet blocks.

More on parameters in Module 2.4.
The KWParamSet block in this example is named “Wangchu”

- **GAUGE**
  The name of the gauge block to which these parameters apply

- **TH**
  Number of grid cells needed to flow into a cell for it to be part of a channel

- **UNDER**
  The interflow flow speed multiplier

- **LEAKI**
  Amount of water leaked from interflow reservoir in each time step

- **ISU**
  Initial value of the interflow reservoir

- **ALPHA** and **BETA**
  Used in the equation \( \text{Streamflow} = \alpha \times (\text{cross-sectional channel area})^\beta \)

- **ALPHA0**
  The alpha value used for overland, not channel, routing
The Task block in this example is named “RunWangchu”

- **STYLE**
  This tells the model what time of task it will be completing (SIMU is used for basic simulations)

- **MODEL**
  Which water balance component will be used (CREST, SAC, or HyMOD)

- **ROUTING**
  Whether to use kinematic wave or linear reservoir routing (KW; LR)

- **BASIN**
  Name of the basin block over which the model should run

```plaintext
[Task RunWangchu]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Wangchu
PREcip=TRMM
PET=FEWSNET
OUTPUT=output\nPARAM_SET=Wangchu
ROUTING_PARAM_SET=Wangchu
TImESTEP=1d
TIME_BEGIN=200101010000
TIME_WARMEND=200102010000
TIME_END=200212310000
```
The Task block in this example is named “RunWangchu”

- **PRECIP**
  PrecipForcing block name

- **PET**
  PETForcing block name

- **OUTPUT**
  The location where the model output will be stored

- **PARAM_SET**
  Name of the CrestParamSet block

- **ROUTING_PARAM_SET**
  Name of the KWParamSet block

```
[Task RunWangchu]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Wangchu
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\ PARAM_SET=Wangchu
ROUTING_PARAM_Set=Wangchu
Timestep=1d
Time_Begin=200101010000
Time_WarmEnd=200101010000
Time_End=200212310000
```
The Task block in this example is named “RunWangchu”

- **TIMESTEP**
  Time step used in the simulation (can be in years, months, days, hours, minutes, or seconds)

- **TIME_BEGIN**
  Time at which the model starts, in YYYYMMDDHHUUSS

- **TIME_WARMEND**
  Optional, time when the warm-up period ends

- **TIME_END**
  Time when the model run ends

```plaintext
[Task RunWangchu]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Wangchu
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\ 
PARAM_SET=Wangchu
ROUTING_PARAM_SET=Wangchu
TIMESTEP=1d
TIME_BEGIN=200101010000
TIME_WARMEND=200102010000
TIME_END=200212310000
```
We’ve so far just described the basic options and requirements in the EF5 control file, but the model has many additional components that will be considered in the rest of the training.

Now let’s discuss two important modeling concepts:

- Model warm-up
- Model states
Warming up the Model

Recall the bucket metaphor from Module 1.2…
And now imagine a time before the model has started running. How do we know how much water is in the buckets?
The simple answer is we usually don’t know.

So we give the model some guess or estimate of the initial conditions or initial model states.

These estimates aren’t perfect, but that usually doesn’t matter, thanks to the process of model warm-up.
Here’s how it works:

- You assign initial conditions to the model using parameters.
- So, let’s say you tell the model that your surface water, groundwater, runoff, and interflow buckets are half full.
- Hopefully that’s more accurate than starting the model completely dry (that is, with no water in any of the buckets).
- Now you let the model run for some specified warm-up period until the bucket levels get closer to reality.

And then you only use the model results for the time period after the warm-up is complete.
Without knowing it, we had a short (1 month) warm-up period in our first example

- The simulation began 1 Jan 2001
- The warm-up period ended 1 Feb 2001
EF5 does not output the results from the model during the warm-up period, because they would not typically be used for any analysis.

Let’s see what happens without a warm-up period.

In `\EF5_training\examples \wangchu\` open `control.txt` and add a `#` symbol before “TIME_WARMEND”
Warming up the Model

This turns that line into a “comment” so that EF5 doesn’t read it as part of the control file anymore.

Save the control file and double-click RunEF5.bat.

Open the results in your output folder.

Notice how small the “Discharge” values are in January 2001 compared to the “Observed”.
I went ahead and created a hydrograph using the instructions in Module 1.2

Our model results (blue) are too low at the beginning of each hydrograph.
What if I start the simulation during the rainy season, with no warm-up period? Let’s try a start date of August 1, 2001 (200108010000)
Warming up the Model

Then let’s zoom in on the beginning of the hydrograph…
You can see the simulation start to pick up between August 12^{th} and 18^{th}, but before then it's far too low.

This suggests a warm-up period of 2 weeks is required, and a month would be pretty safe.

Running without a warm-up period risks the inclusion of bad results at the beginning of your simulation.
Now let’s imagine a different situation

Say I’m running the model every day at the same time on Wang Chu Basin as new rainfall data becomes available.

But I don’t want to waste my computer time starting the model over at the beginning (including a warm-up period) each day just so I can get the latest day’s results.

The solution is to save the model states – the stream flow and the soil moisture at the end of the model run.
Saving model states is a powerful technique.

A large basin with high-resolution information might require a 6 month warm-up period, which could take 6 hours to run.

If I want twice-daily stream flow forecast updates, that means my computer might run 12 hours a day just to provide those two forecasts.
BUT – if I save model states at the end of each forecast I produce, I can run the model forward without a warm-up period, because I know how much water was in all the buckets at the end of the run.

Now my forecast update might only take 5 minutes to run, instead of 6 hours.
This technique allows the experimental OU/NASA forecasts of streamflow and soil moisture in Namibia, over the Okavango River Basin upstream of Rundu.

Each new set of forecast rainfall or estimated satellite rainfall can be “added” to be previous model run without warm-up, so that the results are available quickly.
Recall Example 1

We produced this hydrograph...

How do we know if this is a “good” result from our model?
In general, the result is better the closer the match between the observed streamflow (black) and the discharge (red).

But could you tell if the model was better in 2001 or 2002 just by looking at the hydrograph?

That’s why several model evaluation indices exist.
Bias

Bias can be thought of as a shift up or down in a hydrograph. The lower-right hydrograph now exhibits high bias – the red line has shifted up. This is a conditional bias: the higher the original value, the greater the bias.
Bias

Bias could be constant, or non-stationary
Bias is expressed as a percent
When bias > 0, the modeled values are too high, on average
We like bias values close to zero
How to calculate bias:

\[
\text{Bias} = \left[ \frac{\sum_{i=1}^{n} R_{\text{sim},i} - \sum_{i=1}^{n} R_{\text{obs},i}}{\sum_{i=1}^{n} R_{\text{obs},i}} \right] \times 100
\]

The first term \( \sum_{i=1}^{n} R_{\text{sim},i} \) is the sum of all the simulated streamflow values.

The second term \( \sum_{i=1}^{n} R_{\text{obs},i} \) is the sum of all the observed streamflow values.
Model Evaluation Indices

In Microsoft Excel:

Use the SUM formula to add all the simulated values together

$$\sum_{i=1}^{n} R_{sim,i}$$

Use the SUM formula to add all the observed values together

$$\sum_{i=1}^{n} R_{obs,i}$$

Subtract the sum of observed values from the simulated sum

$$\sum_{i=1}^{n} R_{sim,i} - \sum_{i=1}^{n} R_{obs,i}$$
In Microsoft Excel:

\[
\text{Bias} = \left( \frac{\sum_{i=1}^{n} R_{sim,i} - \sum_{i=1}^{n} R_{obs,i}}{\sum_{i=1}^{n} R_{obs,i}} \right) \times 100
\]

Thus, the bias in this example is 32%
Model Evaluation Indices

**Correlation coefficient**

This measures the agreement between the simulation and the observation time series.

Measured on a 0 to 1 scale, where 1 is desirable and 0 is not desirable.
Correlation coefficient

If bias identifies how far vertically displaced the simulation is from the observations, then correlation coefficient measures horizontal displacement between the two time series. This is highly simplified, but for our purposes it will work.
How to calculate correlation coefficient

It looks complicated, but is actually extremely easy in Microsoft Excel

Just use the CORREL function – our answer is 0.85 for this example
Nash-Sutcliffe Coefficient of Model Efficiency

Referred to as NSCE or “Nash”

Exists on a scale from \(-\infty\) to 1

When NSCE > 0, the model has more skill than if you just used the average stream flow over the period of the simulation

On a hydrograph, NSCE can be visualized as the amount of space or distance between the simulation and the observation, in a general sense

\[
NSCE = 1 - \frac{\sum_{i=1}^{n} (R_{obs,i} - R_{sim,i})^2}{\sum_{i=1}^{n} (R_{obs,i} - \bar{R}_{obs})^2}
\]
Model Evaluation Indices

How to calculate NSCE

Let’s get $R_{obs}$ first

\[
NSCE = 1 - \frac{\sum_{i=1}^{n} \left( R_{obs,i} - R_{sim,i} \right)^2}{\sum_{i=1}^{n} \left( R_{obs,i} - \overline{R}_{obs} \right)^2}
\]

What I’ve done here is take the average of the whole column of observations and then fill all of Column I with that same number
How to calculate NSCE

- Now we’ll calculate the numerator \( \sum_{i=1}^{n} (R_{obs,i} - R_{sim,i})^2 \)
- We need the square of the sum of the differences between each observation and simulation
- Luckily, Excel has a built-in formula called SUMXMY2 that will do this for us
How to calculate NSCE

- Now we’ll calculate the denominator \( \sum_{i=1}^{n} \left( R_{obs,i} - \overline{R_{obs}} \right)^2 \)
- We need the square of the sum of the differences between each observation and the mean observation
- Use SUMXMY2 again
How to calculate NSCE

- Now we’ll put it all together

\[
NSCE = 1 - \frac{\sum_{i=1}^{n}(R_{obs,i} - R_{sim,i})^2}{\sum_{i=1}^{n}(R_{obs,i} - \bar{R}_{obs})^2}
\]

- Take 1 minus the numerator over the denominator

<table>
<thead>
<tr>
<th>D</th>
<th>E</th>
<th>Formula Bar</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precip(mm) PET(mm) h SM(%)</td>
<td>Fast Flow(Slow Flow) Average of observations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.31</td>
<td>0</td>
<td>0.0001</td>
<td>99.34386266</td>
<td>Sum of Observed Values: 69441.36</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.41</td>
<td>0</td>
<td>0.0003</td>
<td>99.34386266</td>
<td>Sum of Simulated Values: 91544.53</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.46</td>
<td>0</td>
<td>0.0001</td>
<td>99.34386266</td>
<td>Subtract: 22103.17</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.48</td>
<td>0</td>
<td>0</td>
<td>99.34386266</td>
<td>Divide: 0.3183</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.49</td>
<td>0</td>
<td>0</td>
<td>99.34386266</td>
<td>Multiply by 100: 31.82998</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.53</td>
<td>0</td>
<td>0.0001</td>
<td>99.34386266</td>
<td>Correlation Coefficient: 0.853987</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.53</td>
<td>0</td>
<td>0</td>
<td>99.34386266</td>
<td>NSCE Numerator: 3170080</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.58</td>
<td>0</td>
<td>0.0001</td>
<td>99.34386266</td>
<td>NSCE Denominator: 6209129</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>25.67</td>
<td>0</td>
<td>0.0003</td>
<td>99.34386266</td>
<td>NSCE: 0.489449</td>
<td></td>
</tr>
</tbody>
</table>
Back to Our Question…

Was the model output “good”?

The correlation coefficient (0.85) is very good
Remember this is a 0-1 scale

The bias (32%) is definitely higher than we’d prefer
We want values close to 0

The NSCE (0.49) is adequate
A really well-calibrated model should have a NSCE of 0.7 or more, if we are lucky

The verdict: more calibration could be definitely done, but this is a pretty good hydrograph

![Graph showing streamflow and precipitation over time]

Legend:
- Green line: Precipitation [mm h$^{-1}$]
- Red line: Discharge [m$^3$ s$^{-1}$]
- Black line: Observed [m$^3$ s$^{-1}$]
Coming Up….

The next module is

DEM Derivatives

You can find it in your \EF5_training\presentations directory

Module 2.1 References

EF5 v0.2 Readme, (March 2015).