EF5

Automatic Calibration

University of Oklahoma/HyDROS

Module 2.4

Outline – Day 2

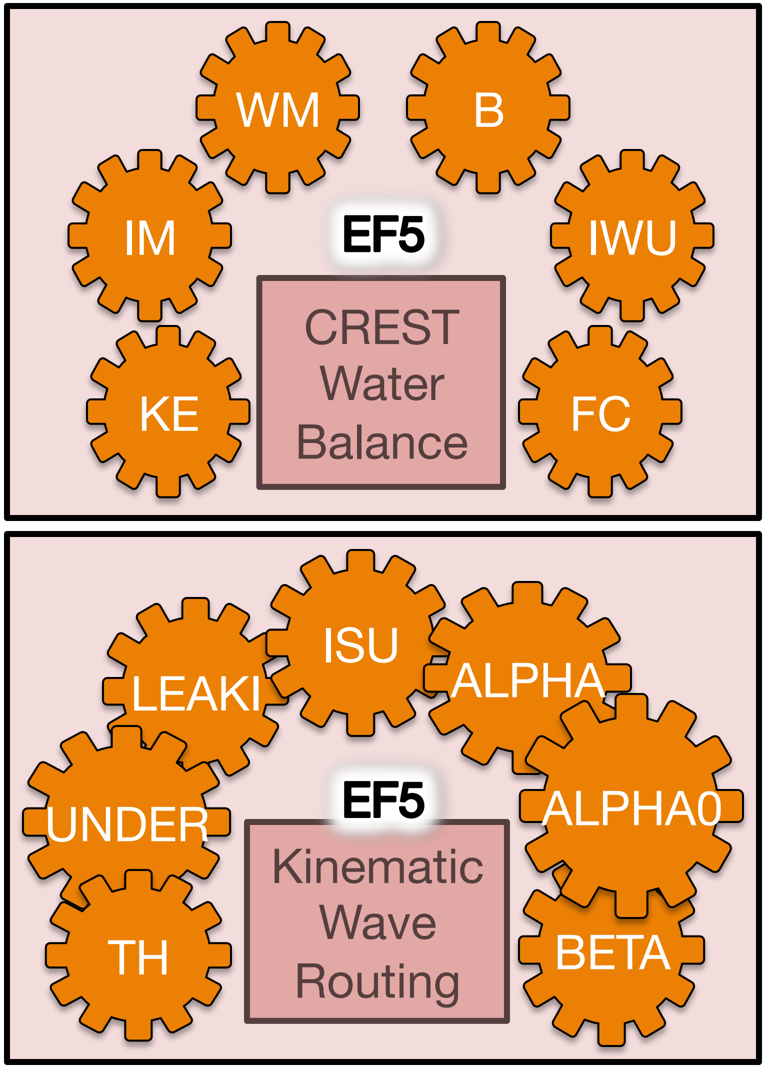
EF5 OVERVIEW

DEM DERIVATIVES

RAINFALL AND PET

AUTOMATIC CALIBRATION

- Description of EF5/CREST parameters
- Description of EF5/KW parameters
- Automatic calibration algorithms
- Calibrate EF5 for Example 2
- Calibration and validation periods



Calibration

What is calibration?

Calibration is the process of adjusting model parameters so that the model more accurately simulates stream flow (when compared to observed stream flow)

We adjust each parameter and see how the model evaluation indices (NSCE, CC, and bias) change

This can be a lengthy and complex process – there is no one right answer when it comes to calibration

EF5 Parameters

First, let's review what we know about EF5's parameters

There are a total of 13 of them

They're divided into 2 categories

- The CREST parameters, which affect the water balance (how water is partitioned and accounted for in each cell)
- The kinematic wave parameters, which affect how water is routed from cell to cell

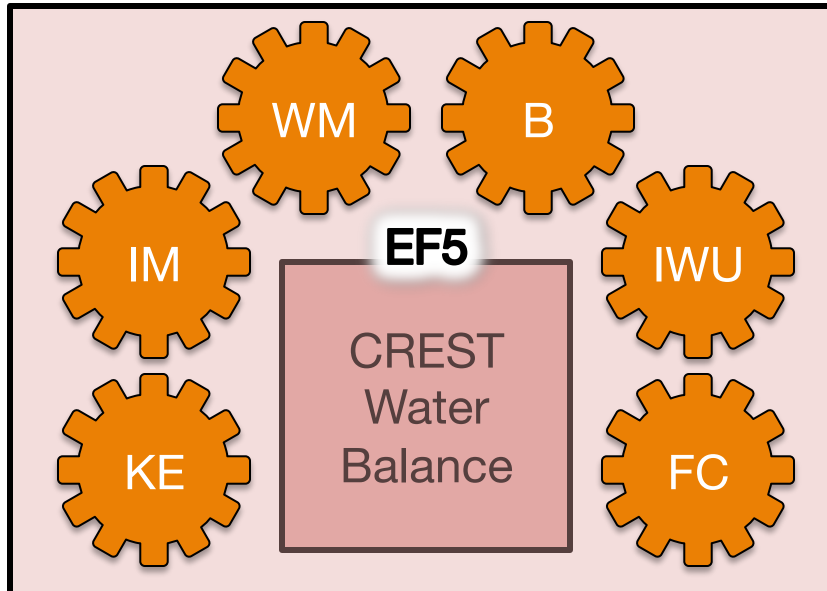
Some parameters affect the model results more than others

WM, B, ALPHA, BETA, and ALPHA0 are generally the most important, but others can be too

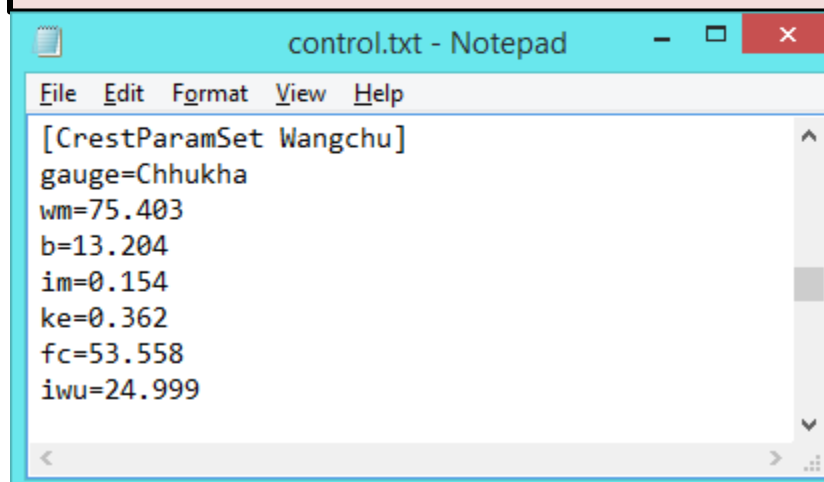
CREST Parameters

The CrestParamSet block contains the six water balance parameters

- WM
  Maximum soil water capacity
- PB
  The exponent of the variable infiltration curve
- 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



Wang Chu (Example 1) parameters →




```
[CrestParamSet Wangchu]
gauge=Chhukha
wm=75.403
b=13.204
im=0.154
ke=0.362
fc=53.558
iwu=24.999
```

WM

WM is the maximum soil water capacity (depth integrated pore space) of the soil layer in the model, in millimeters

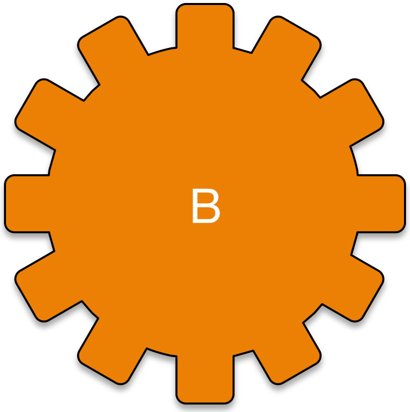
- This is how much water the soil layer can store
- Physically, this a function of several soil properties
- If I increase WM, that means there's more space in the soil for water, which means less runoff will be produced

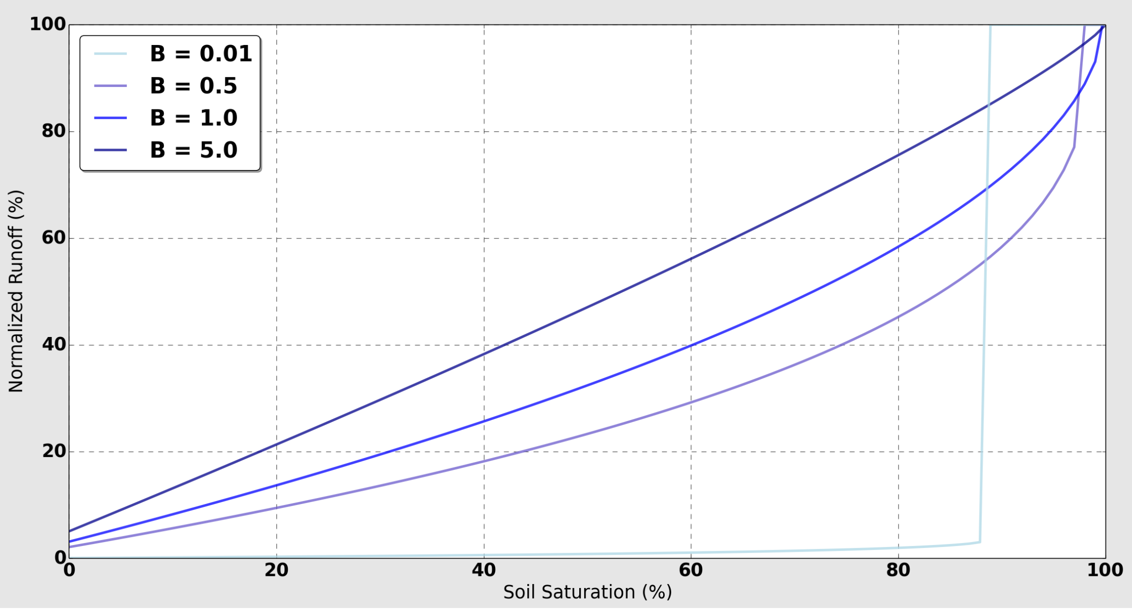
WM will generally be between 5.0 and 250.0 mm

B is the exponent of the variable infiltration curve (VIC)

Remember the VIC governs how much water enters the soil layer and how much remains at the surface as runoff

If I increase B, I tend to produce more runoff





B can range between 0.1 and 20.0

IM

IM is the impervious area ratio

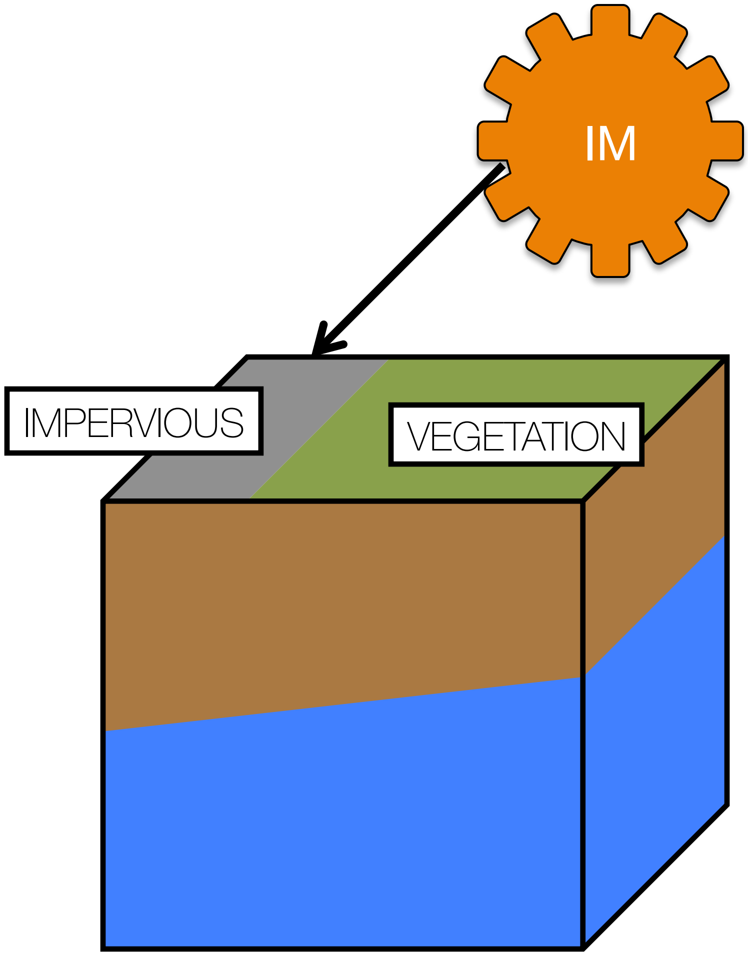
You can think of this as the percentage area of your modeled domain covered in roofs, concrete, rocky soils and other impervious materials

If I increase IM, my runoff increases

IM can range from nearly 0 to 0.5

You generally do NOT want to manually calibrate IM, because it is fairly easy to observe from surveys or remote sensing

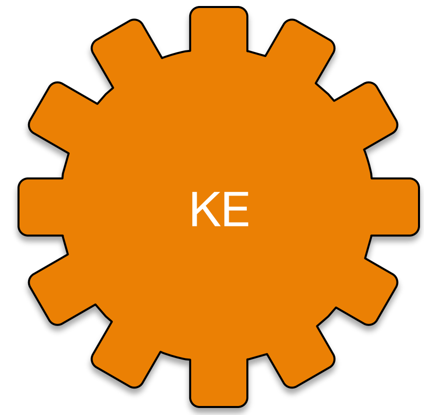
So if you know 10% of a basin is covered in rocks or concrete, your IM should be about 0.10



KE

KE is the multiplier to convert between input PET and local actual ET

This is essentially an adjustment to the monthly global PET grid to make it more accurately reflect conditions over the modeled basin



The PET forcing provided in this training course tends to be a little too "hot"

In other words, too much water gets evaporated when the original PET grid values are used

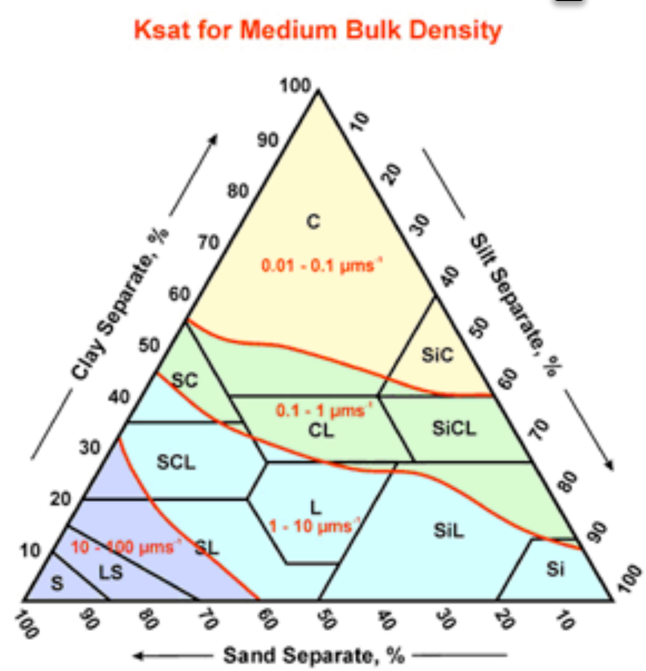
KE can range from 0.001 (one one-thousandth of the PET grid) to 1.0 (the entire PET grid)

You can use values over 1.0, but this worsens the already "hot" tendency of the provided PET grids

FC

FC is the soil saturated hydraulic conductivity (K_{sat}) in mm/hr

- This describes how easily water moves through saturated soil
- The higher the value, the more easily water can travel through saturated soils
- Higher values tend to decrease runoff
- FC can be determined from soil properties and field measurements, which can help reduce the possible range of values in the calibration process



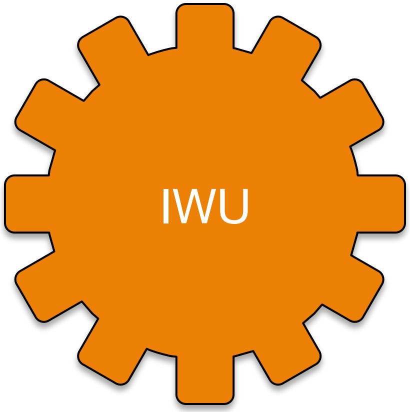
Guide for estimating K_{sat} from soil, from NSSH Part 618 (Subpart B), NRCS, USDA

FC can range from 0.0 to 150.0

IWU

IWU is the initial value of soil water

- It is a percentage of WM
- If you have a long enough warm-up period in your simulation, the value of this parameter is unimportant
- Even if you don't have a warm up period, IWU can be safely estimated around 25.0
- This is because the soil isn't bone-dry (0.0) but it's also a safe bet that it's probably not totally saturated either



CREST Parameter Summary

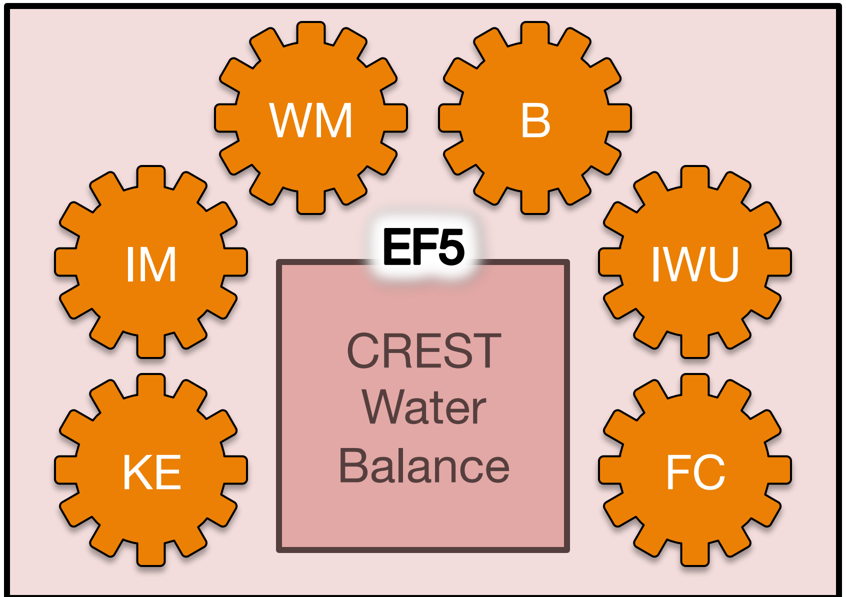
To summarize:

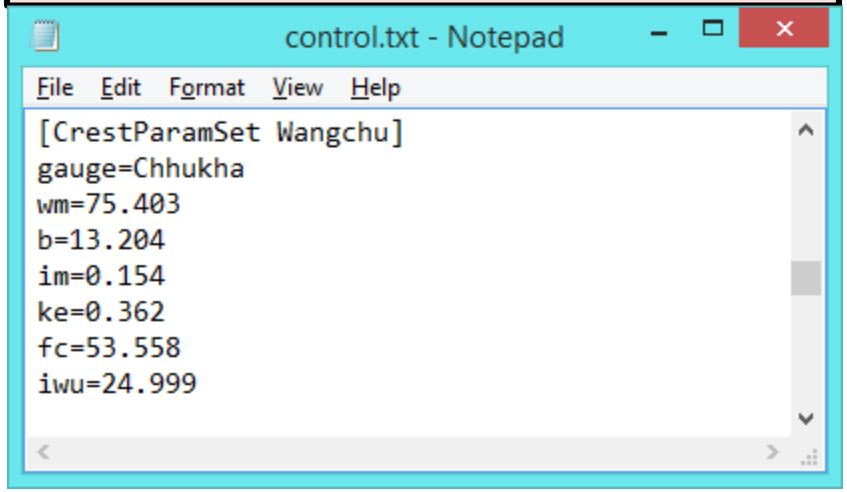
WM and B are important to the accuracy of the simulation, so focus your attention here when manually calibrating

FC and IM can be measured or estimated for a basin via other means, so pick values for these and leave them alone

IWU is not important if you use a warm-up period

KE should generally be less than 1.0 when using the PET grids provided with this training



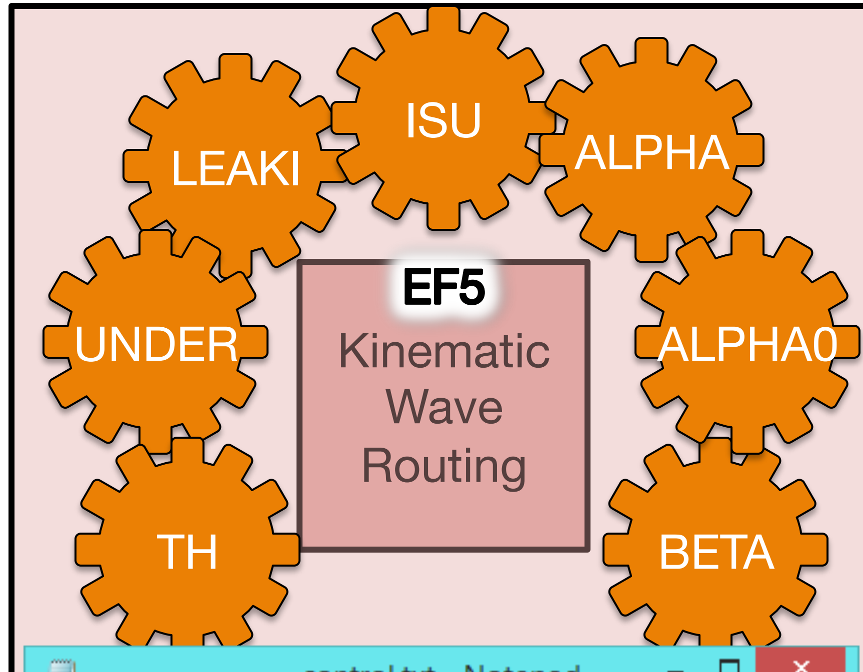


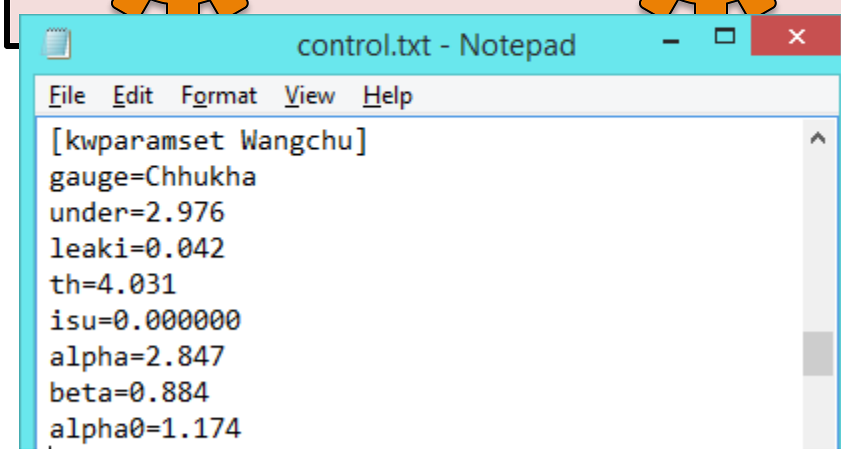
Routing Parameters

The KWParamSet block contains the seven routing parameters in EF5

- 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 Streamflow = alpha*(cross-sectional channel area)^{beta}
- ALPHA0
  The alpha value used for overland, not channel, routing

Wang Chu (Example 1) parameters →





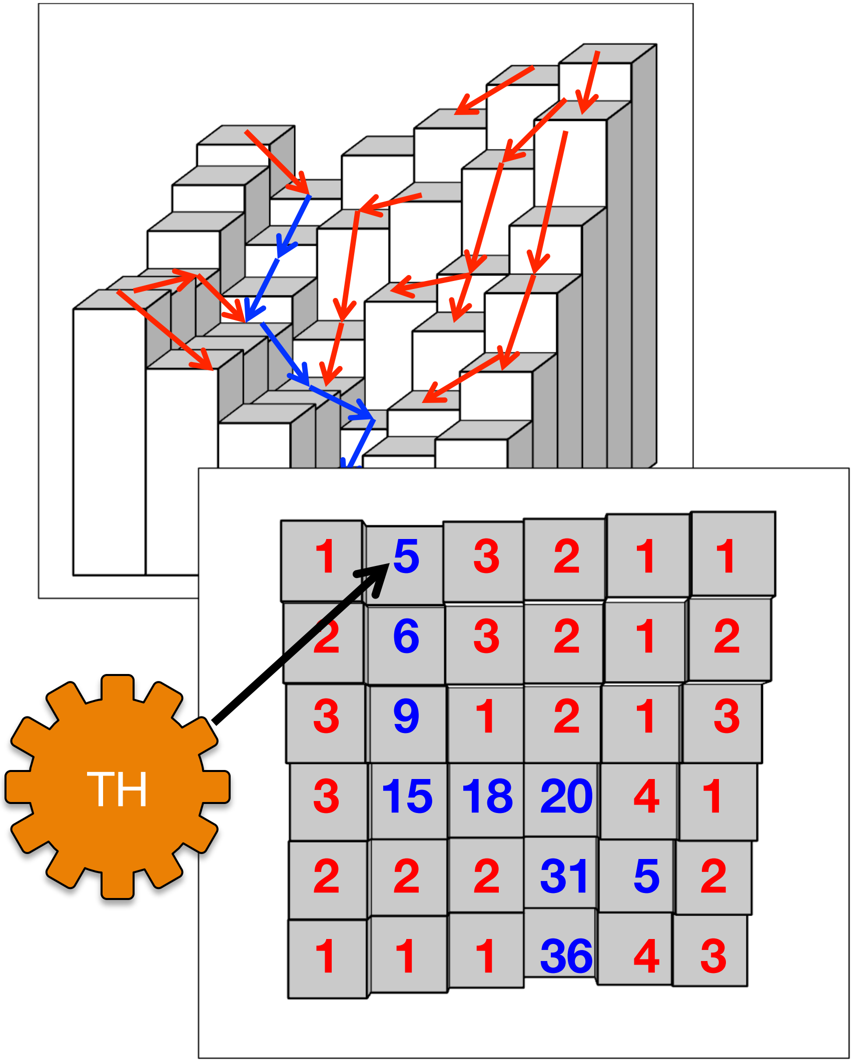

```
[kwparamset Wangchu]
gauge=Chhukha
under=2.976
leaki=0.042
th=4.031
isu=0.000000
alpha=2.847
beta=0.884
alpha0=1.174
```

TH

TH is the threshold for how many cells must drain into a cell for it to become part of a river in the model

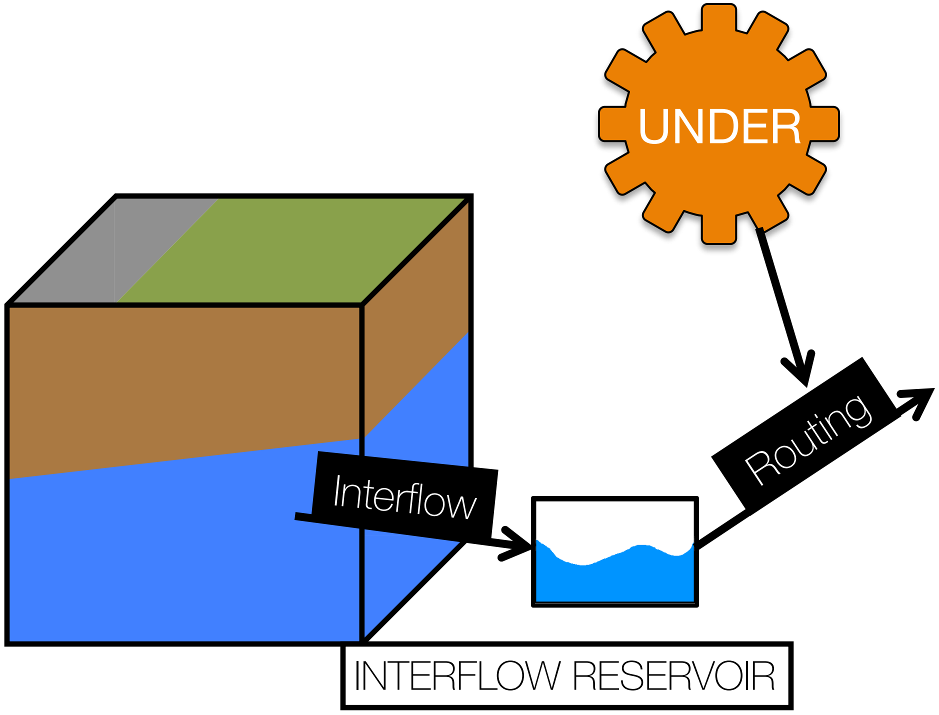
- Determined from the FAC grid
- Dependent on resolution of the topographical files
- As the FAC resolution increases, the value of TH should also increase

If you convert from grid cells to actual area (in km^2), TH should be between 30 and 300 km^2



UNDER is the interflow flow speed multiplier

- Higher values of this mean water moves faster through the soil layer, which can result in faster peaks in a hydrograph
- This only affects the timing of the flood wave, not the volume of water making it into the river channel

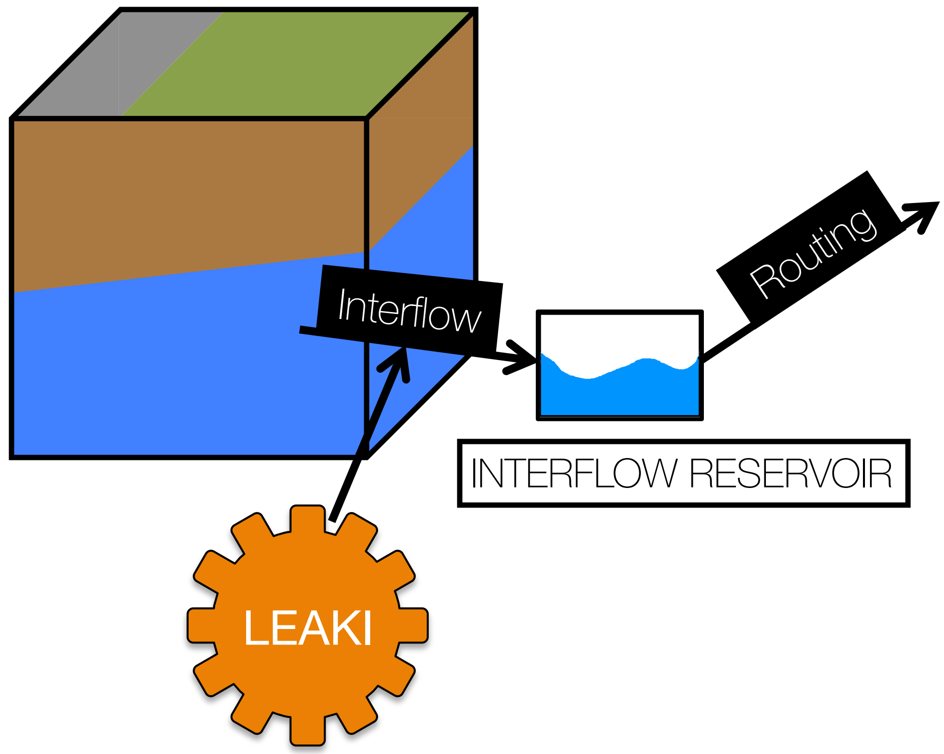


This parameter can range between 0.0001 and 3.0

LEAKI

LEAKI is the amount of water leaking out of the interflow reservoir at each time step

- This is expressed as a percentage of the total water in the interflow reservoir
- The water that leaks out moves on to the next downstream cell's interflow reservoir
- Increasing this parameter will result in faster peaks

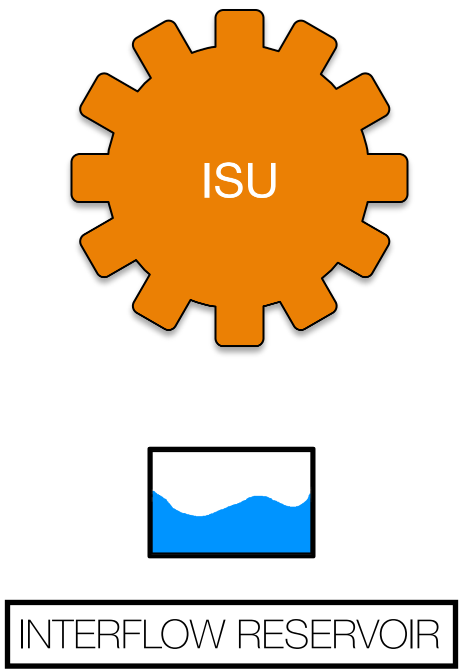


This can range between 0.01 and 1.0

ISU

ISU is the initial value of the interflow reservoir

- If you use a warm-up period this parameter is unnecessary
- Setting this parameter to something other than zero will result in an unrealistic peak in the hydrograph at the very beginning of the simulation time

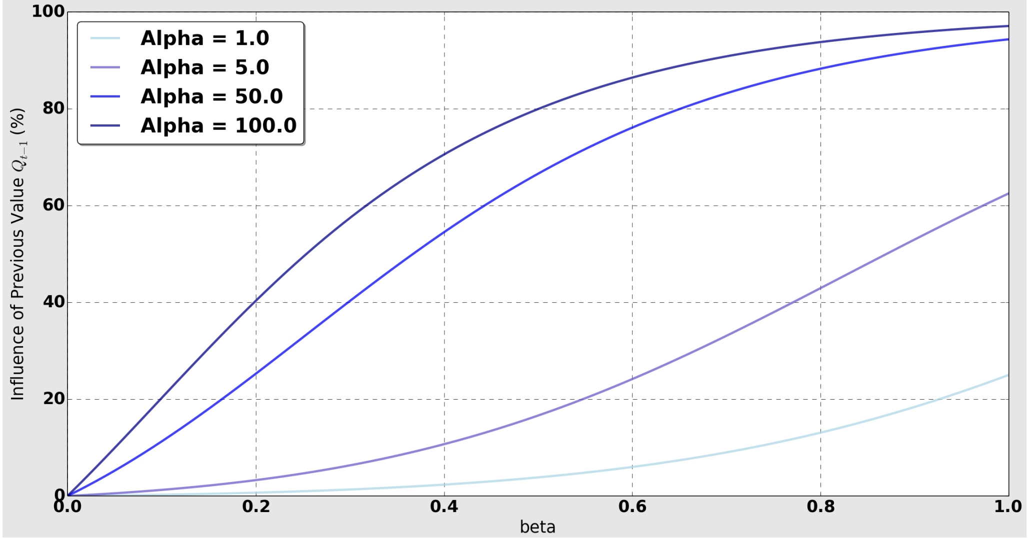


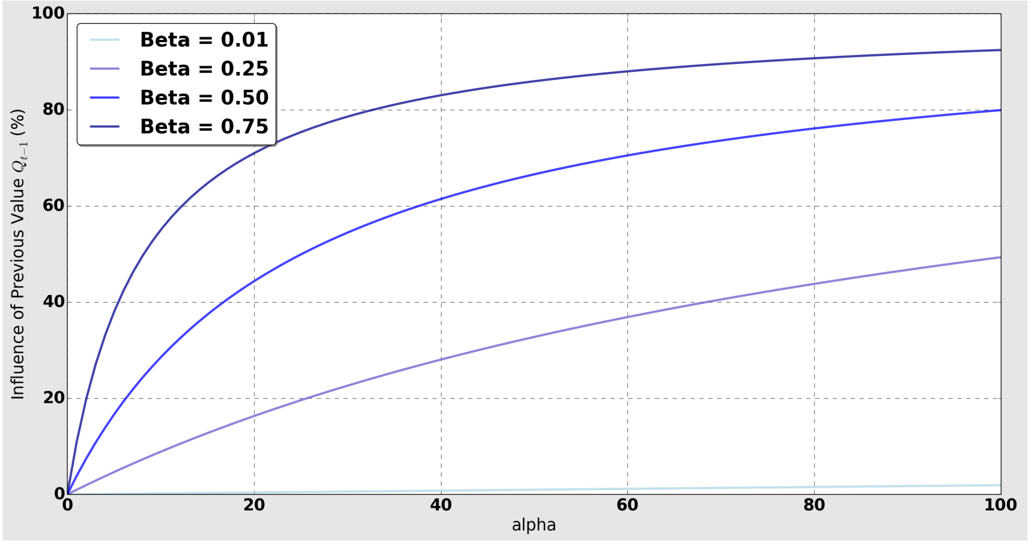
This parameter should usually be set to 0

ALPHA

ALPHA is the multiplier in the equation $Q = \alpha A^{\beta}$

- This governs routing
- Q is stream flow and A is cross-sectional area of the stream channel
- For a constant A, Q, and BETA, increasing ALPHA slows down my flood wave (that is, the hydrograph peak is delayed)





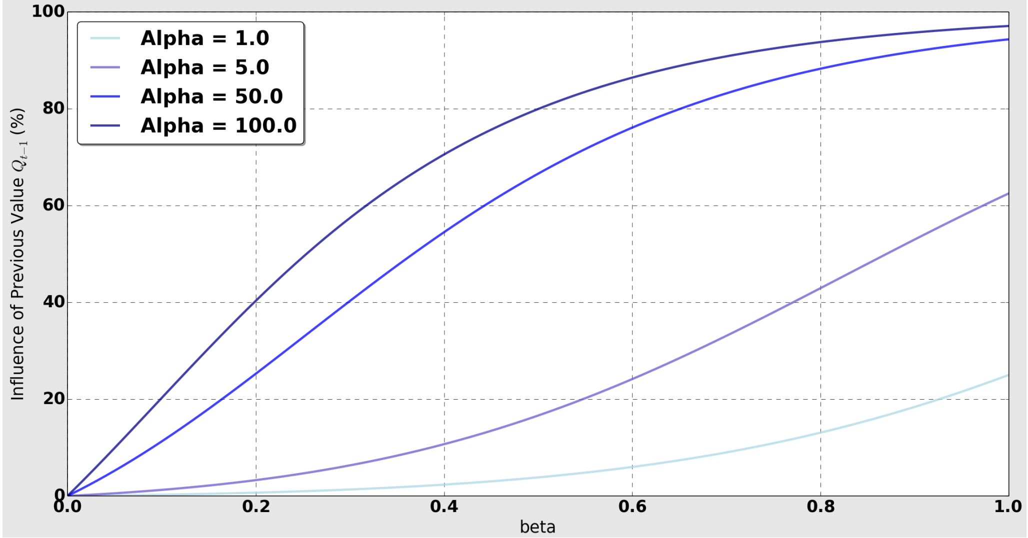
ALPHA can vary between 0.01 and 3.0

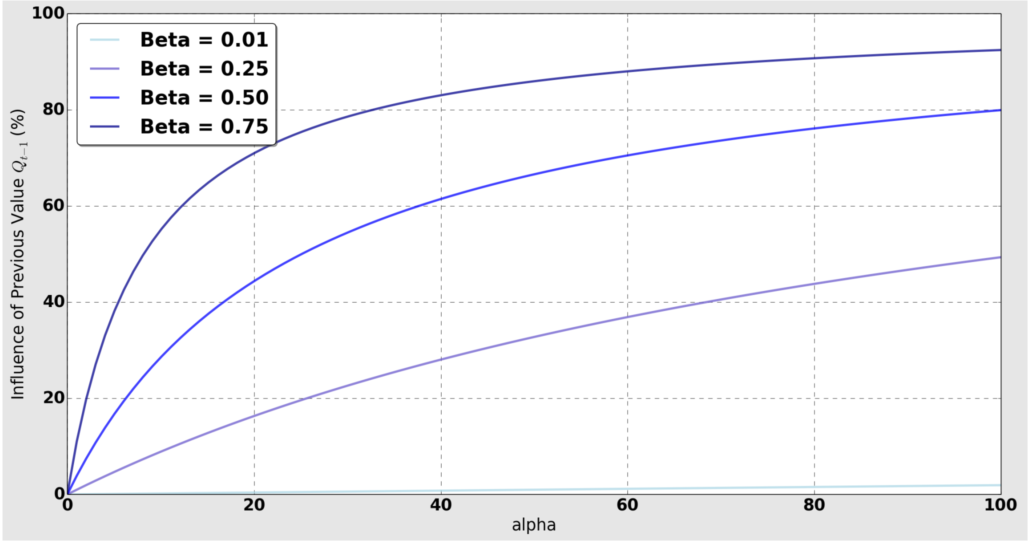
BETA

BETA is the exponent in the equation Q = αA$^{\beta}$

For a constant A, Q, and ALPHA, as I increase BETA, my flood wave slows down

BETA can vary between 0.01 and 1.0





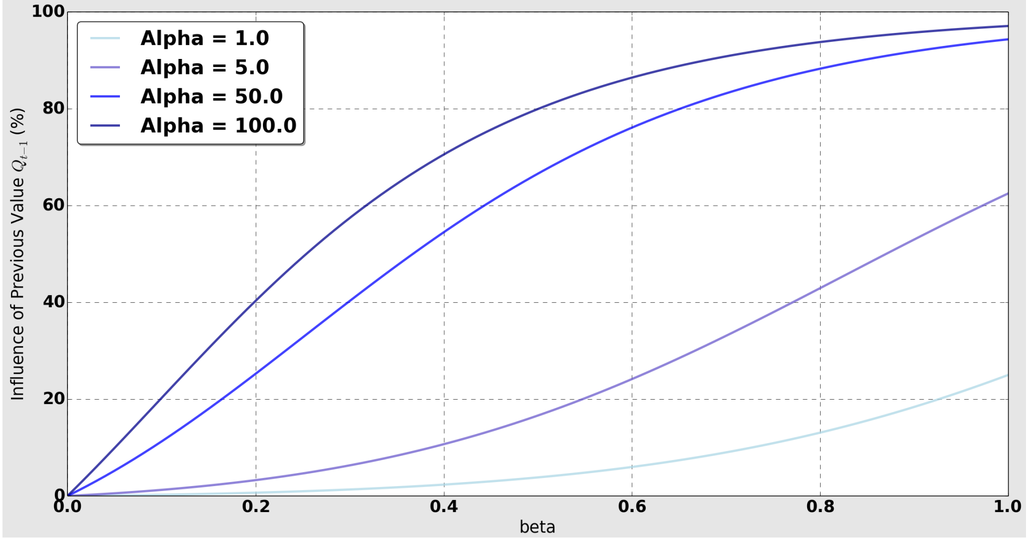
ALPHA0

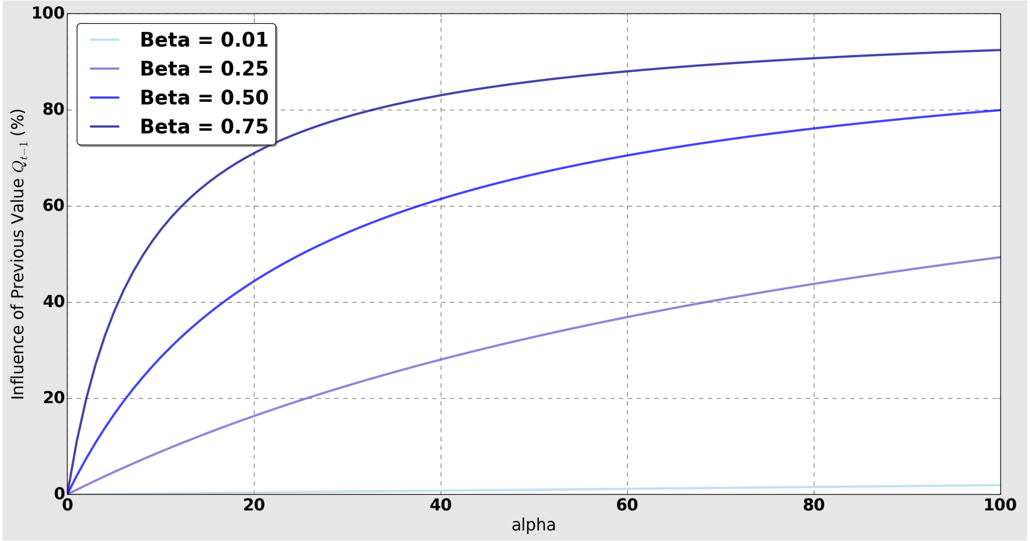
ALPHA0 is the multiplier in the equation $Q = \alpha A^{0.6}$

This governs routing for overland cells

It behaves the same as the channel ALPHA

Note that in overland cells, the BETA value is set at 0.6; this value is found by solving Manning's Equation





ALPHA0 can vary between 0.01 and 5.0

Routing Parameter Summary

In routing calibration, the ALPHA, BETA, and ALPHA0 parameters typically matter the most

- Keep ISU at zero and use a warm-up period
- Increase LEAKI and UNDER to speed up a flood wave
- Increase ALPHA, BETA, and ALPHA0 to slow it down
- Convert the 30-300 km^2 to grid cells, based on your simulation's resolution, and then try TH values in that range

Automatic Calibration

Adjusting these parameters individually is called manual calibration, which is a time consuming process that requires active involvement

How do you know when you have really reached the optimum set of parameters?

Automatic calibration methods solve these problems for us by searching the available parameter space using a single objective function and the hydrological model

DREAM Algorithm

Differential Evolution Adaptive Metropolis (DREAM)

- Global parameter optimization method
- Uses Markov chain Monte Carlo (MCMC) simulations to find the best objective value
- Each simulation is referred to as a "function evaluation" when doing automatic calibration. We want to find the set of parameters, **p**, that gives us the best objective function results
- *Objective_function*(*f*(**p**, **x**)) where *f* is our hydrologic model
- Objective function choices in EF5 include NSCE (Nash-Sutcliffe Coefficient of Model Efficiency), CC (correlation coefficient) and Sum of Squared Errors

DREAM Algorithm

Differential Evolution Adaptive Metropolis (DREAM)

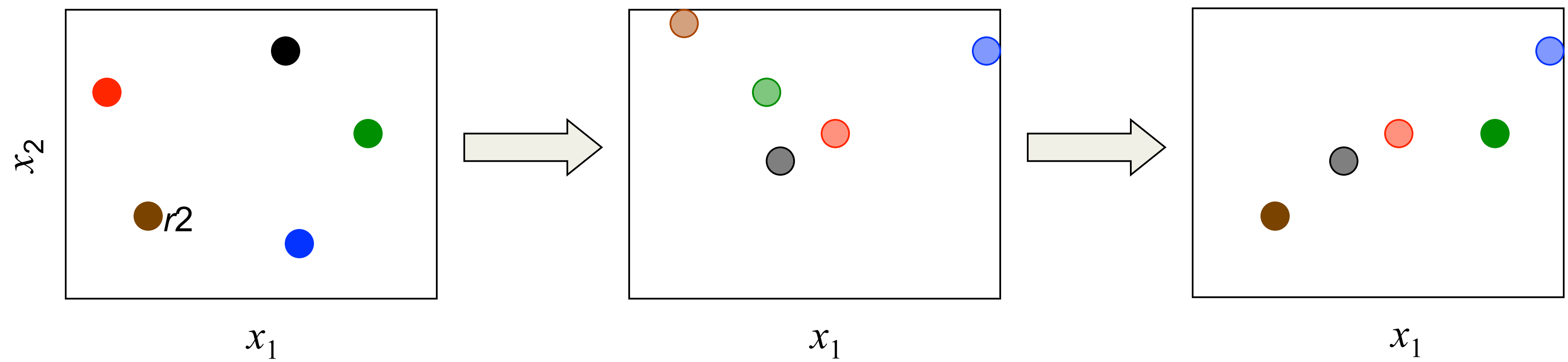
Tunes scale and orientation of proposal parameter distribution using differential evolution (genetic)

Takes full advantage of distributed computer networks

Returns marginal posterior parameter distributions and their covariances (useful for quantifying parametric uncertainty → ensemble prediction)

DREAM Algorithm

I. Initialize *N* different chains — II. Create Proposals — III. Accept / Reject



1. Initialize *N* different Markov Chains in the parameter space
2. Create proposals in each chain *i = 1,…,N* according to:

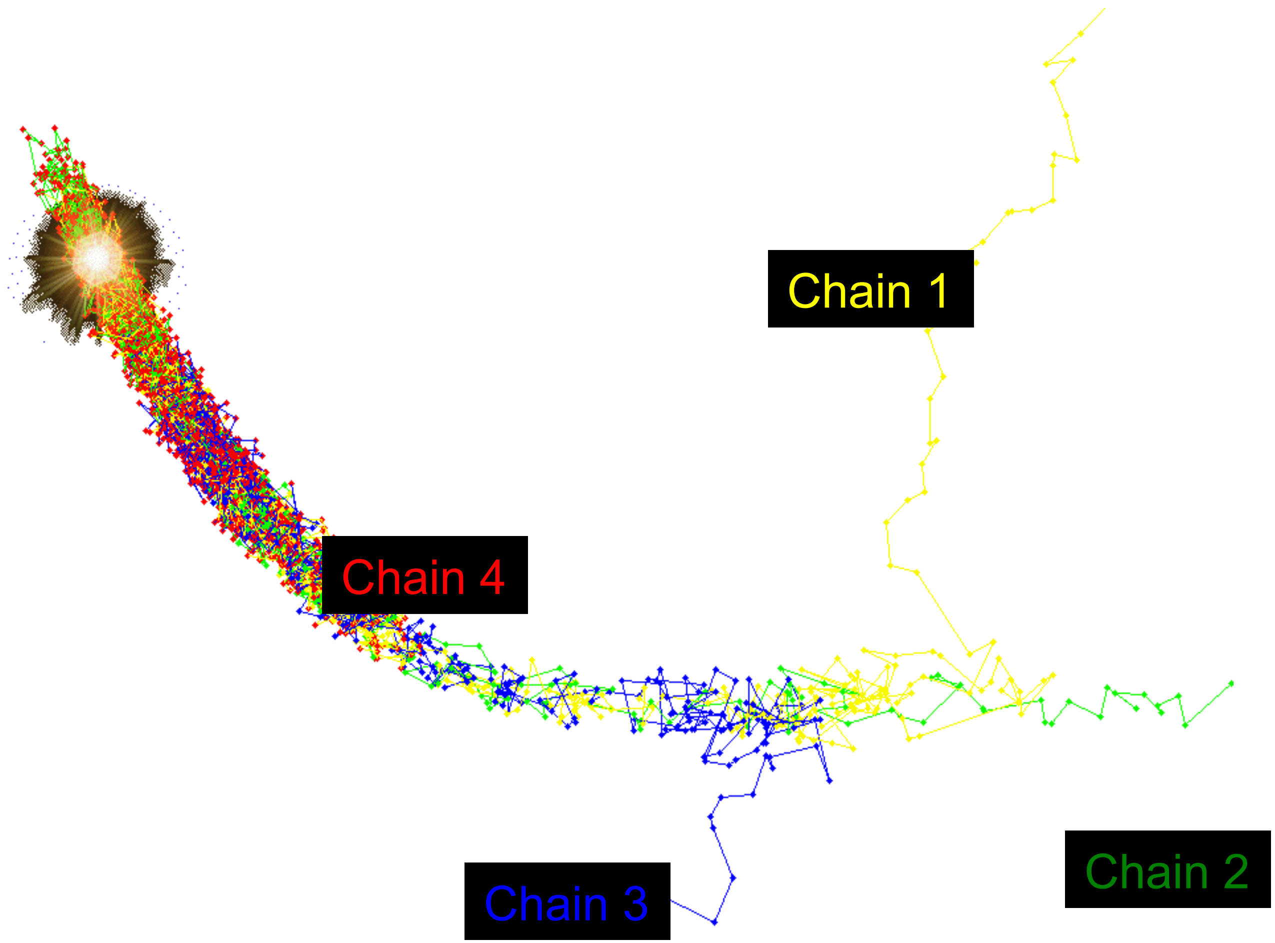
$$\mathbf{z}^i = \mathbf{x}^i_{t-1} + \gamma(\mathbf{x}^{r1}_{t-1} - \mathbf{x}^{r2}_{t-1}) + \mathbf{e}, \quad r1 \neq r2 \neq i \quad ; \quad \gamma = 2.4 / \sqrt{2n}$$

3. Compute the Metropolis probability:

$$\alpha = \min(\pi(\mathbf{z}^i) / \pi(\mathbf{x}^i_{t-1}), 1)$$

4. If α > U[0,1]; $\mathbf{x}^i_t = \mathbf{z}^i$ otherwise remain at the current location

DREAM Algorithm



DREAM Algorithm

Differential Evolution Adaptive Metropolis (DREAM)

- DREAM requires parameter ranges to search, so picking good ranges is very important
  - These ranges are a soft limit when doing manual calibration but a hard limit for automatic calibration methods
- Ranges can't be too big, or you require more function evaluations to properly search the parameter space
- Ranges can't be too small, or the parameter space may not be large enough to yield good objective function results
- Hydrologic model parameters do influence each other, so our problem space is under-determined

Differential Evolution Adaptive Metropolis (DREAM)

Suggested default ranges for CREST and Kinematic Wave parameters:

Parameter	Min	Max
CREST WM	5.00	250.0
CREST B	0.10	20.0
CREST IM	0.01	0.5
CREST FC	0.01	150.0
KW ALPHA	0.01	3.0
KW BETA	0.01	1.0
KW ALPHA0	0.01	5.0

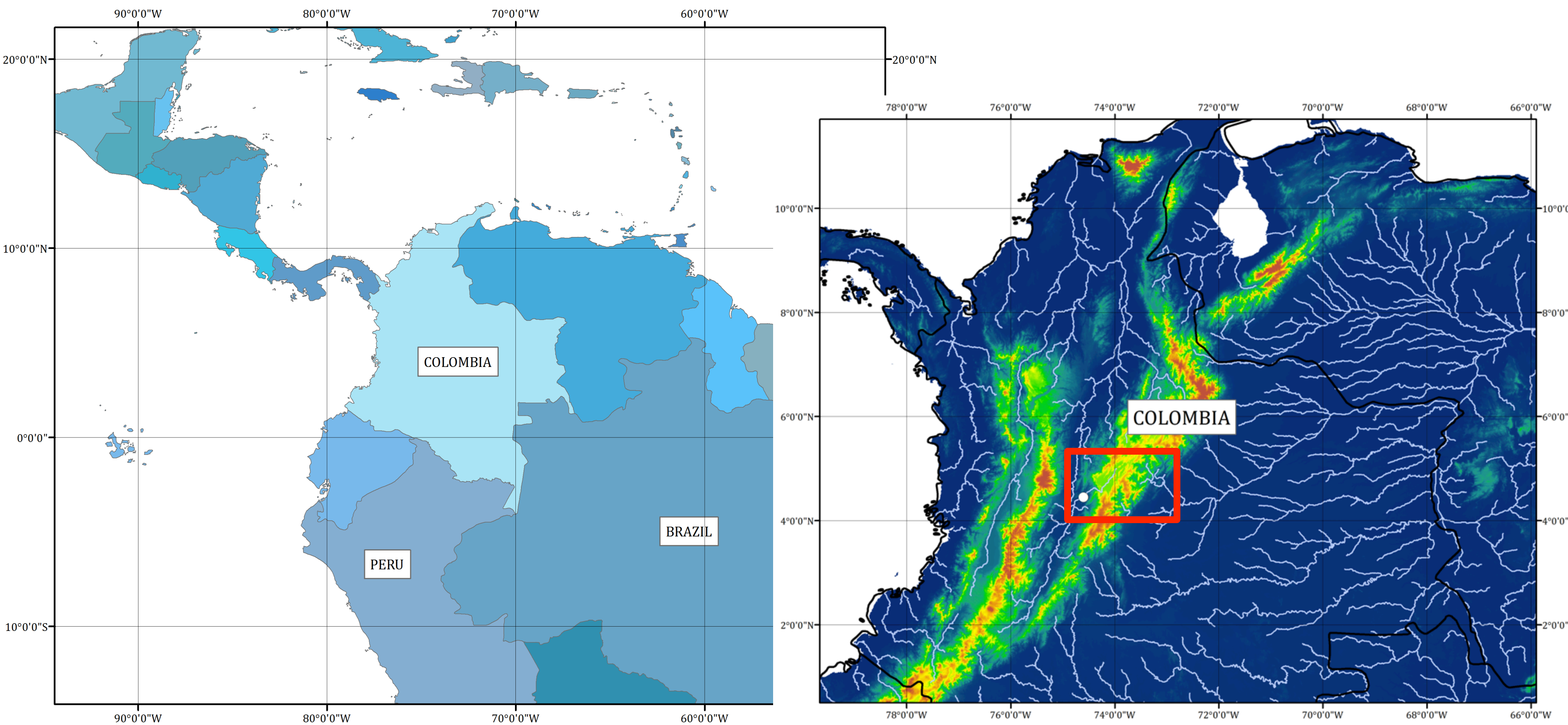
Automatic Calibration Strategies

Use the longest available time period when not conducting research

Start with a small number (100-200) of function evaluations (`dream_ndraw`) and see if calibration is progressing as expected

Example 2

Let's use EF5 in DREAM mode to automatically calibrate Example 2



Control File Preparation

Open `control.txt` in `EF5_training/examples/example2`

So we think our Basic, PrecipForcing, PETForcing, Gauge, and Basin blocks are ready to go

What's left?

We need to tell EF5 about our parameters, instructions for how to calibrate them, and instructions on how to actually run the model

control.txt - Notepad

File Edit Format View Help

```
[Basic]
DEM=basic\dem.tif
DDM=basic\fdr.tif
FAM=basic\fac.tif
PROJ=geographic
ESRIDDM=false
SelfFAM=true

[PrecipForcing TRMM]
TYPE=BIF
UNIT=mm/d
FREQ=d
LOC=precip\
NAME=TR_YYYYMMDD.bif

[PETForcing FEWSNET]
TYPE=BIF
UNIT=mm/d
FREQ=m
LOC=pet\
NAME=PET025.MM.bif

[Gauge PuentePortillo]
LON=-74.6
LAT=4.45
OBS=obs\puente_portillo.csv
BASINAREA=6000.00
OUTPUTTS=TRUE

[Basin Bogota]
GAUGE=PuentePortillo
```

CrestParamSet and KWParamSet

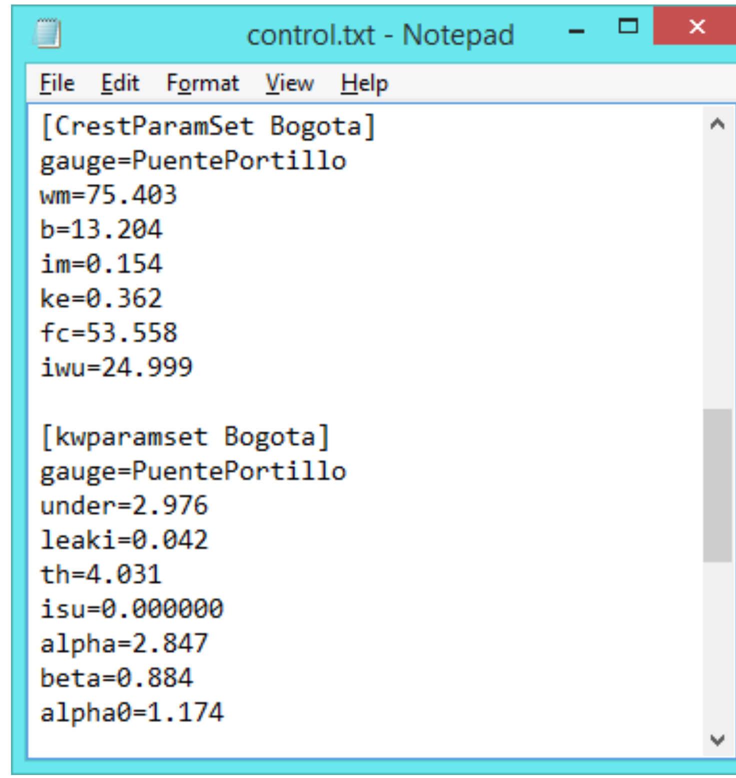
CrestParamSet

- The block name is Bogota (but can be anything you want as long as you use this same name later in the Task block)
- gauge should be the name of the Gauge block for which you want these parameters to apply (in this case, we already picked PuentePortillo as our gauge name)

Let's go ahead and keep the original Wang Chu (Example 1) parameters, since we're going to be calibrating anyway

kwparamset

- The block name is again Bogota (but can be anything you want as long as you reuse this name in the Task block)
- And again, gauge should be the name of the Gauge block for which you want these parameters to apply (in this case, we already picked PuentePortillo as our gauge name




```
[CrestParamSet Bogota]
gauge=PuentePortillo
wm=75.403
b=13.204
im=0.154
ke=0.362
fc=53.558
iwu=24.999

[kwparamset Bogota]
gauge=PuentePortillo
under=2.976
leaki=0.042
th=4.031
isu=0.000000
alpha=2.847
beta=0.884
alpha0=1.174
```

CrestCaliParams and kwcaliparams

CrestCaliParams

- The block name is `Bogota` (but can be anything you want as long as you tell EF5 the name in the Task block)
- `gauge` should be the name of the Gauge block for which you want these parameters to apply (in this case, we already picked `PuentePortillo` as our gauge name)
- `objective` is the objective function we want DREAM to maximize (in this case, we say `nsce` because we want to maximize the NSCE of our simulation)
- `dream_ndraw` is the number of iterations or times we want DREAM to run
- Then the six CREST parameters are listed, where the number before the comma is the minimum end of the parameter range to be tested by DREAM, and the number after is the maximum end of the parameter range

control.txt - Notepad

File Edit Format View Help

```
[CrestCaliParams Bogota]
gauge=PuentePortillo
objective=nsce
dream_ndraw=100
wm=5.0,250.0
b=0.1,20.0
im=0.00999999,0.5
ke=0.001,1.0
fc=0.0,150.0
iwu=24.9999,25.0

[kwcaliparams Bogota]
gauge=PuentePortillo
alpha=0.01,3.0
alpha0=0.01,5.0
beta=0.01,1.0
under=0.0001,3.0
leaki=0.01,1.0
th=1.0,10.0
isu=0.0,0.0000001
```

CrestCaliParams and kwcaliparams

kwcaliparams

- The block name is `Bogota` (but can be anything you want as long as you tell EF5 the name in the Task block)
- `gauge` should be the name of the Gauge block for which you want these parameters to apply (in this case, we already picked `PuentePortillo` as our gauge name)
- Then the seven routing parameters are listed, where the number before the comma is the minimum end of the parameter range to be tested by DREAM, and the number after is the maximum end of the parameter range

control.txt - Notepad

```
[CrestCaliParams Bogota]
gauge=PuentePortillo
objective=nsce
dream_ndraw=100
wm=5.0,250.0
b=0.1,20.0
im=0.00999999,0.5
ke=0.001,1.0
fc=0.0,150.0
iwu=24.9999,25.0

[kwcaliparams Bogota]
gauge=PuentePortillo
alpha=0.01,3.0
alpha0=0.01,5.0
beta=0.01,1.0
under=0.0001,3.0
leaki=0.01,1.0
th=1.0,10.0
isu=0.0,0.0000001
```

Task Block

The task block tells EF5 what to do with the topographical, forcing, parameter, and location information we've given it

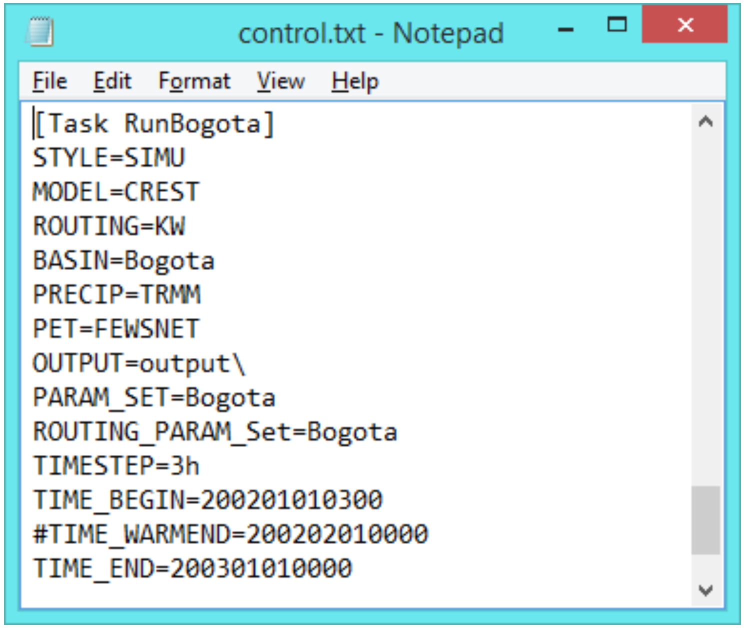
- The name here is `RunBogota` but it can be anything you want, as long as it matches the Execute block (which we'll talk about in later slides)
- `STYLE` tells EF5 what you want to do, in a general sense, so `SIMU` means you want to do a regular simulation and `cali_dream` means you want to calibrate
- `MODEL` tells EF5 which water balance model core you want to use; here we'll stick with `CREST`
- `ROUTING` tells EF5 which routing model you want to use; here we'll stick with `KW`, short for kinematic wave
- `BASIN` tells EF5 which basin, or collection of gauges, you want to run. The name doesn't matter as long as it matches with a basin name as specified in the Basin block earlier in the control file, which would be `Bogota`, in this example

control.txt - Notepad

```
[Task RunBogota]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000
```

Task Block

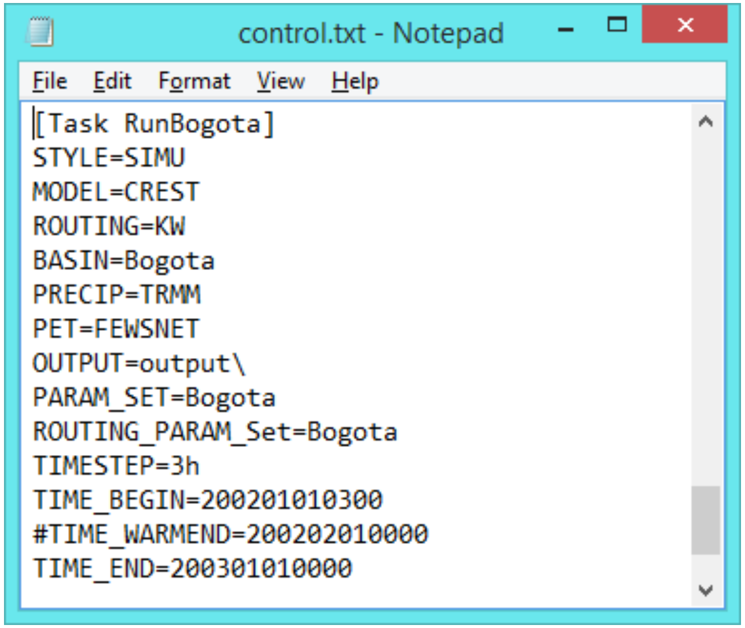
- PRECIP tells EF5 which PrecipForcing block (from earlier in the control file) you want to use; here we have only one PrecipForcing block: TRMM
- PET works the same way, our only PETForcing block is called FEWSNET
- OUTPUT is the file path, relative to the location of control.txt, where you want to have EF5 write the model output
- PARAM_SET and ROUTING_PARAM_SET tell EF5 which parameter sets you want to use; here we only have one of each and they're both named Bogota
- TIMESTEP tells EF5 how often to run in time, and since our precipitation is available every three hours, we'll use 3h


control.txt - Notepad
File Edit Format View Help

```
[Task RunBogota]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000
```

Task Block

- `TIME_BEGIN` is the date and time when the simulation will be begin (this should correspond to a time when precipitation will be available). Let's go ahead and start our simulation at the beginning of our precipitation, January 1, 2002 at 03 UTC, or `200201010300`, in YYYYMMDDHHUU format
- `TIME_WARMEND` is the date and time when the "warm-up" period of the model ends. Here, it appears with `#` in front, which means it has been "commented" out, so EF5 would ignore it
- Finally, `TIME_END` is the date and time when the simulation ends. Let's go ahead and end our simulation at the end of our precipitation, January 1, 2003 at 00 UTC, or `200301010000`, in YYYYMMDDHHUU format

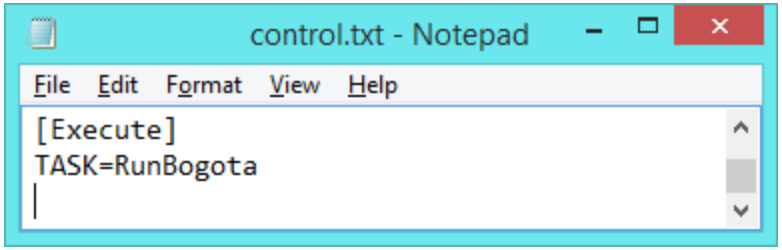



```
[Task RunBogota]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000
```

Execute Block

The Execute block is the final piece of the puzzle

- `TASK` tells EF5 which task block to activate, which would be `RunBogota`, in this example



Now we're finally ready to run this example…

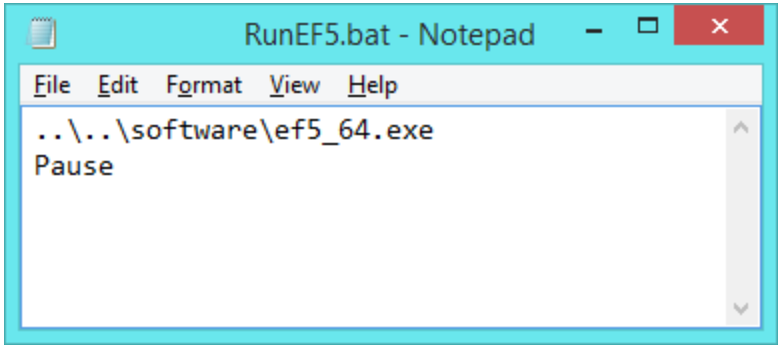
Create a Batch File

Right-click in some blank space in your `EF5_training/examples/example2` folder, and select "New" and "Text Document"

Name it `RunEF5.txt`

Open it and type

```
..\..\software\ef5_64.exe
Pause
```

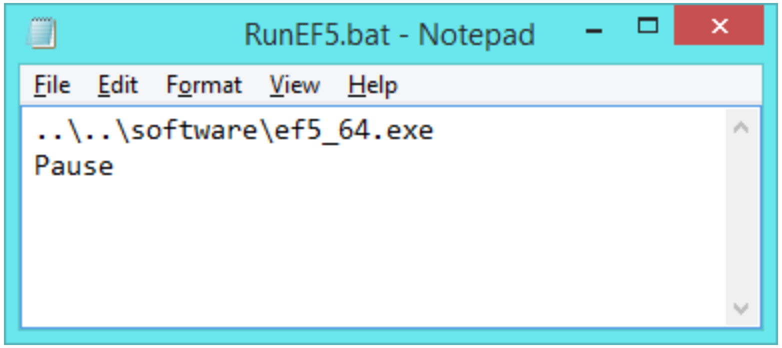


(If you're running on a 32-bit computer, type

```
..\..\software\ef5_32.exe
Pause
```

instead)

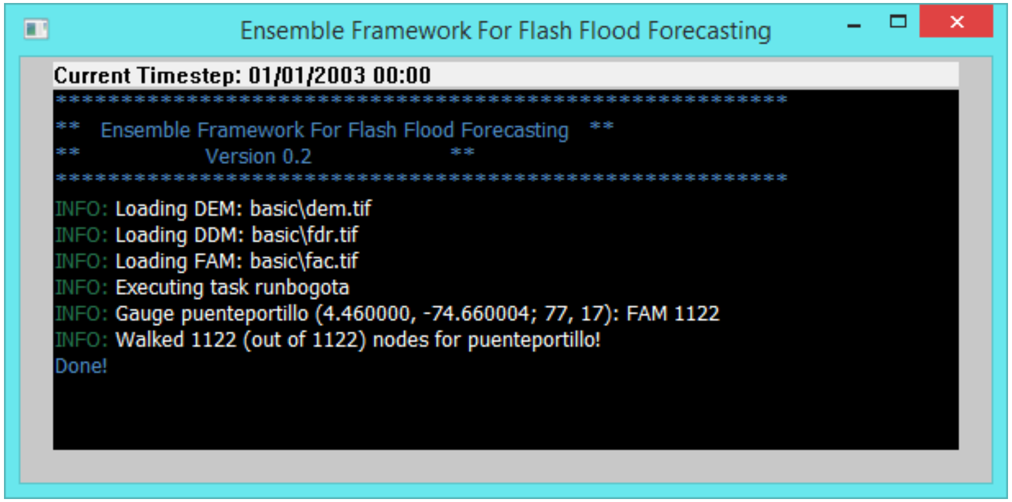
Using the "File" menu, go to "Save As…" and then in the "Save as type:" dropdown box select "All Files (*.*)"



Save it as `RunEF5.bat`

Double-click `RunEF5.bat`

Wait for EF5 to finish →



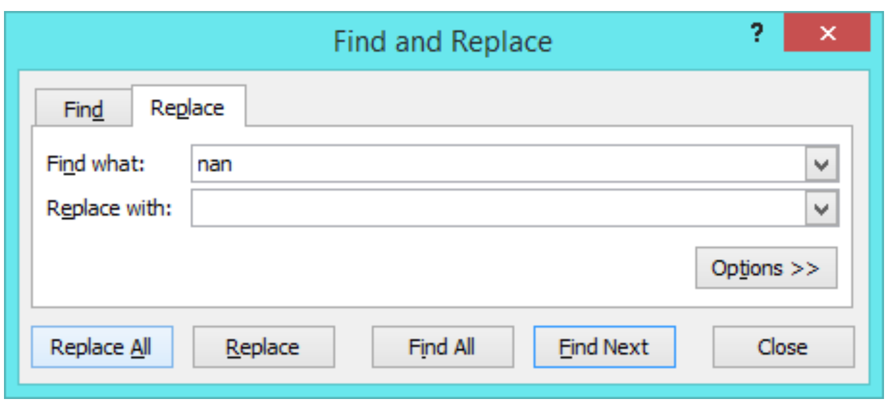
Assessing the Result

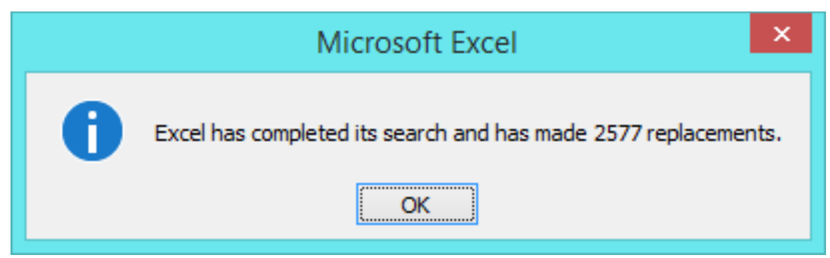
Your `EF5_training\examples\example2\output` **folder should contain a file named** `ts.puenteportillo.crest.csv`

Open this in Microsoft Excel

Use CTRL+F to bring up “Find and Replace” and click the “Replace” tab

Find “nan” and replace it with nothing and then click “Replace All”, click “OK” on the popup and then “Close” in the “Find and Replace” dialog box





Assessing the Result

Start with CC

- Use CORREL and compare discharge (Column B, rows 2 to 2920) to observations (Column C, rows 2 to 2920)
- You should get something around 0.45

Then let's find the bias

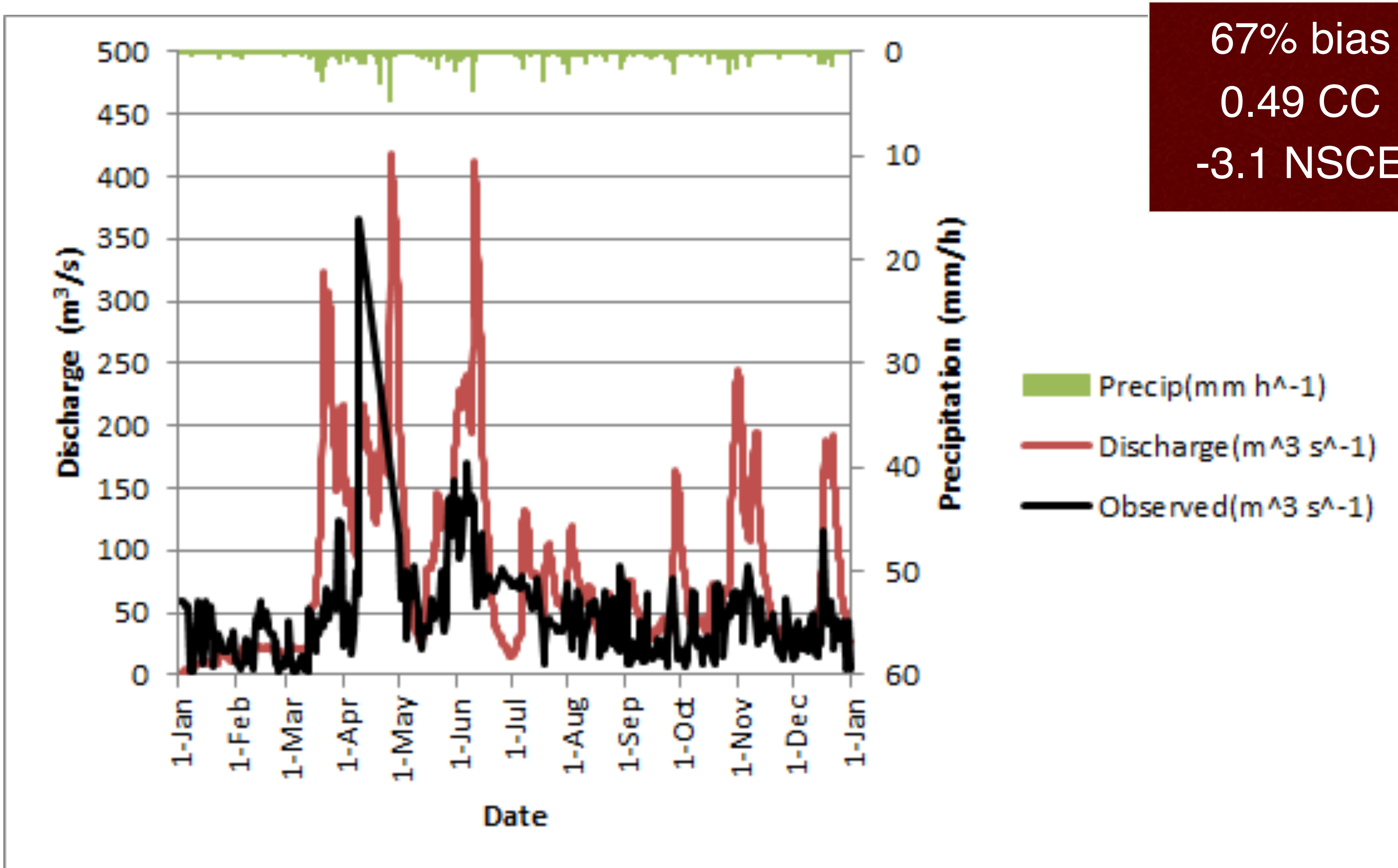
- Add all the simulation values together
    - Use this formula: =SUMIF(C2:C2920,"<>"&"",B2:B290)
- Add all the observed values together (15,850)
- Subtract observed values from simulated values (10,608)
- Divide by the sum of the observed values (0.669)
- And multiply by 100 (66.9%)

Assessing the Result

Finally let's look at NSCE

- Take the average of Column C (46.34573) and fill all of Column I with this same value
- Use SUMXMY2 with Column C as the first argument and Column B as the second (1,716,009)
- Now use SUMXMY2 with Column C as the first argument and Column I as the second (420,797)
- Divide the first SUMXMY2 by the second, and subtract from 1 (-3.08)

Assessing the Result



67% bias

0.49 CC

-3.1 NSCE

Time to Calibrate

Let's calibrate the model for this simulation

We already have calibration parameter blocks in our control file

Now we just need a second Task block to tell EF5 to calibrate

Second Task Block

Copy the original Task block and paste below it to create a second Task block, as shown at right

control.txt - Notepad

File Edit Format View Help

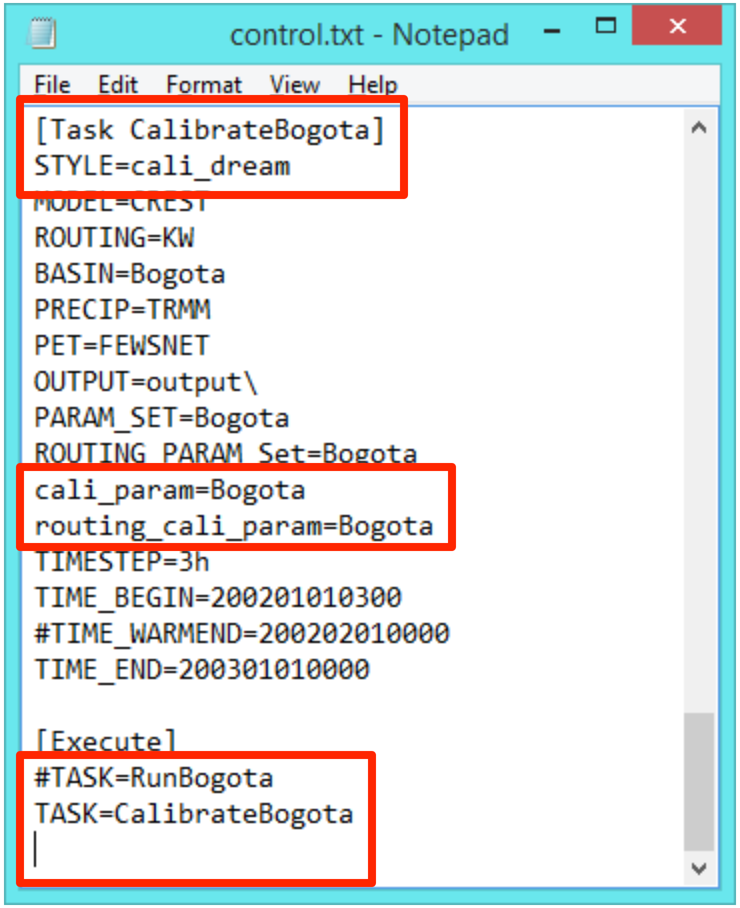
```
[Task RunBogota]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000

[Task RunBogota]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000
```

Task Block

Now let's start editing the second Task block

- First, rename it to `CalibrateBogota`
- Change `STYLE` to `cali_dream`
- Add two new lines below `ROUTING_PARAM_Set` and call them `cali_param` and `routing_cali_param`
- Name both of them `Bogota`


control.txt - Notepad

File Edit Format View Help

```
[Task CalibrateBogota]
STYLE=cali_dream
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
cali_param=Bogota
routing_cali_param=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000

[Execute]
#TASK=RunBogota
TASK=CalibrateBogota
```

And add a new line to your Execute block

- `TASK=CalibrateBogota`
- Comment out (`#`) the `TASK=RunBogota` line
- Finally, make sure there is a blank space after the last line of your Execute block

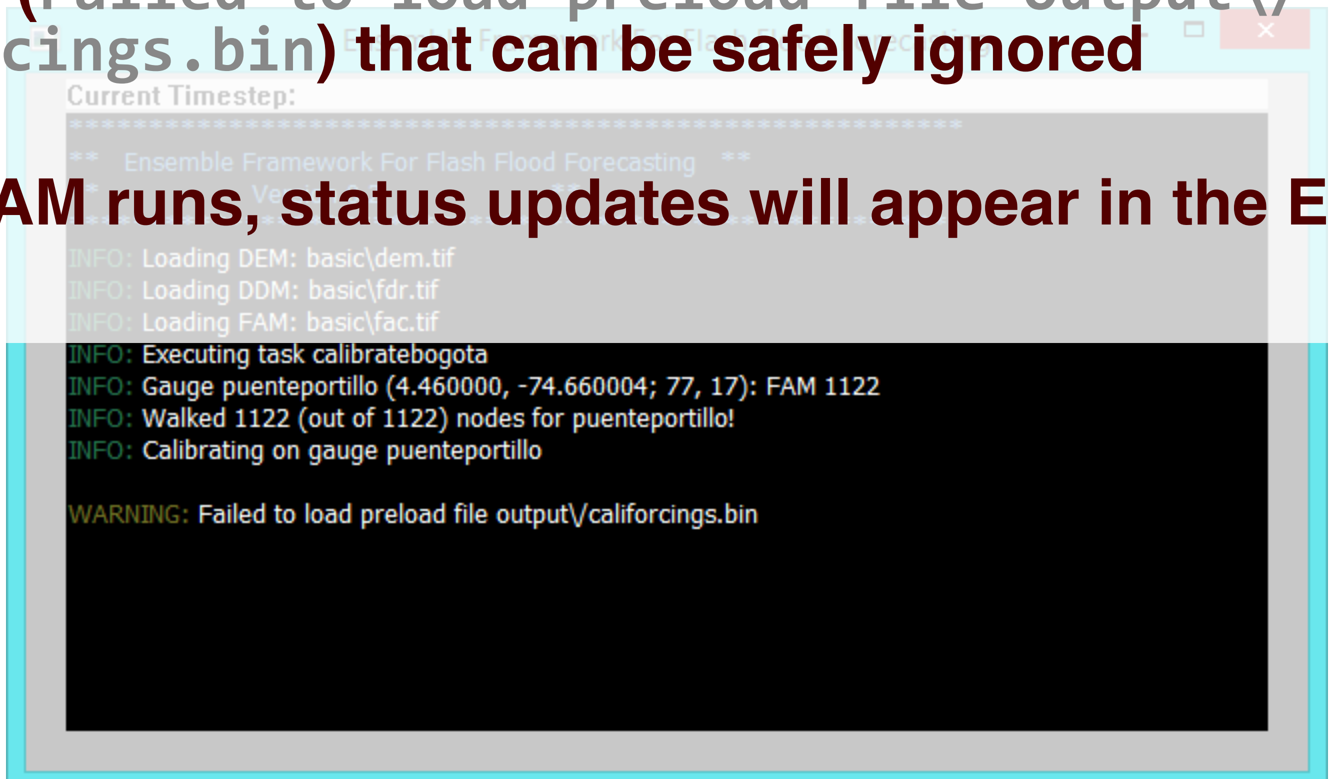
Save your control file

Running DREAM

Double-click `RunEF5.bat`

The first time you calibrate on a new example, you'll get a warning (`Failed to load preload file output\/califorcings.bin`**) that can be safely ignored**

As DREAM runs, status updates will appear in the EF5 window



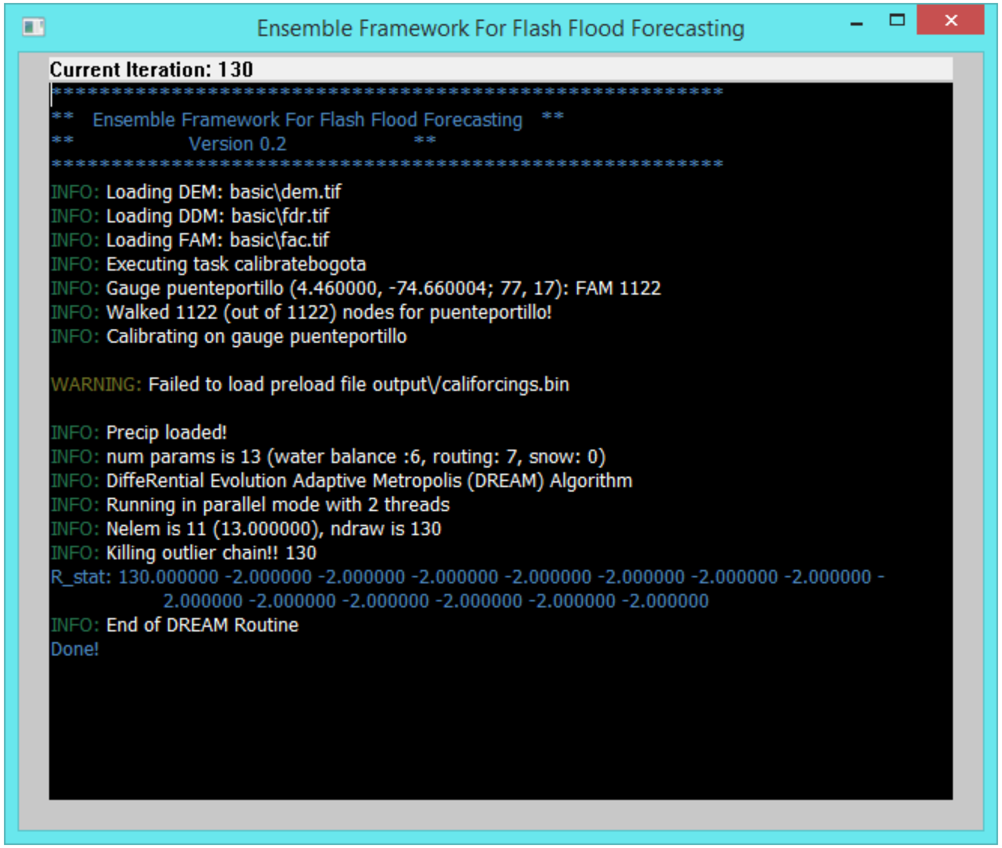
Running DREAM

The white bar at the top of the window tells you how far along in the process DREAM is (it will display the current iteration as a multiple of 13)

Remember that we set our `dream_ndraw` to 100

DREAM automatically converts this to a multiple of 13 (130, which you see at the top of the window)

As in simulation mode, you will see `Done!` at the end of a successful calibration



Checking DREAM Output

Once you see `Done!`**, check** `EF5_training\examples\example2\output\cali_dream.puenteportillo.crest.csv` **for the result**

The second-to-last column – N – shows the NSCE value at each iteration

Remember, DREAM includes randomness, so everyone will get different answers, but my NSCE climbed to 0.1983 (in Excel, check cell N117)

Your new parameters are printed at the bottom of the output spreadsheet (and mine appear at right of the slide)

Copy these to the clipboard, and then exit the file (you can save it if you want so that you can look at it later)

[WaterBalance]
wm=137.524994
b=10.948049
im=0.223177
ke=0.937125
fc=78.484680
iwu=24.999907
[Routing]
under=2.280746
leaki=0.014993
th=4.660986
isu=0.000000
alpha=0.511733
beta=0.819149
alpha0=4.499268

Re-Run EF5

Paste your new calibrated parameters into the `CRESTParamSet` and `kwParamSet` blocks of your control file

Then remove the comment (`#`) from the `RunBogota` line of the control file and comment the `CalibrateBogota` line of the control file

Save the control file

Run EF5

Assessing the Result

You can follow the instructions from Modules 2.1 and 2.2 to open up the EF5 output from this calibrated simulation and calculate the skill statistics and produce a hydrograph

My CC climbed to 0.57 from 0.45

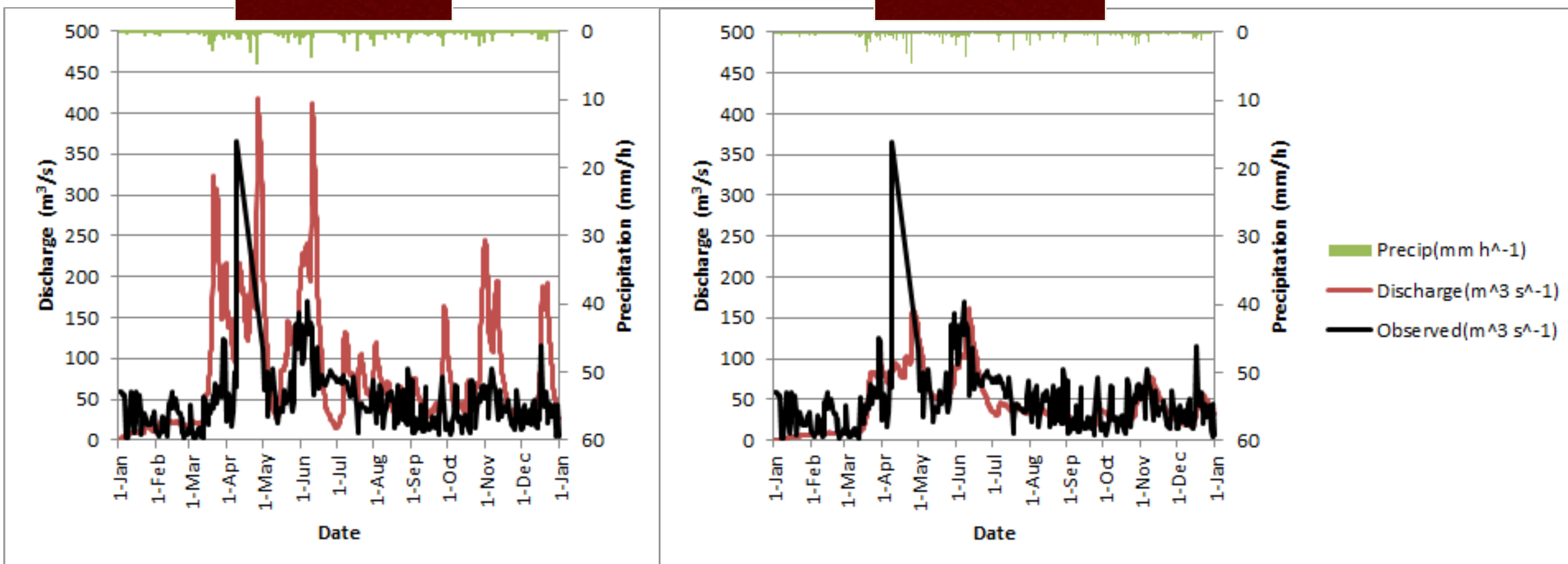
My NSCE (we already knew this from DREAM) climbed from -3.1 to 0.20)

My bias improved to -14% from 67%

The new hydrograph is at lower right and the old hydrograph is at lower left

67% bias
0.49 CC
-3.1 NSCE

-14% bias
0.56 CC
0.20 NSCE

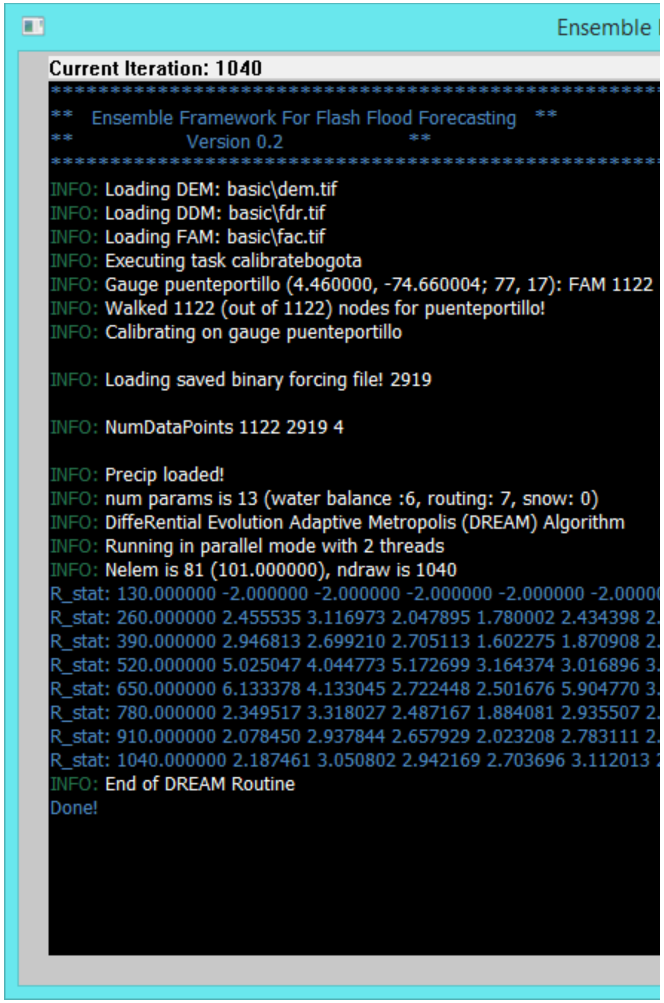


Running DREAM

We can improve our NSCE by calibrating even further

Usually this means setting the `dream_ndraw` **to a higher value, like** `1000`

Now run EF5 again (don't forget to switch out your Execute block!) and then open your calibration output once the routine finishes (and note how this takes a lot longer [~30 min on my computer] with a higher value of `dream_ndraw`**!)**



Checking DREAM Output

Compare this set of results to the first ones

Which of your parameters changed?

What is your new NSCE value?

This time, my NSCE got to 0.33, which means I could try a higher value of `dream_ndraw` or some other techniques, which we'll discuss in Module 3.1

Old	New
[WaterBalance]	[WaterBalance]
wm=137.524994	wm=164.967010
b=10.948049	b=14.707238
im=0.223177	im=0.166324
ke=0.937125	ke=0.787010
fc=78.484680	fc=29.390564
iwu=24.999907	iwu=24.999998
[Routing]	[Routing]
under=2.280746	under=1.273950
leaki=0.014993	leaki=0.011370
th=4.660986	th=9.473580
isu=0.000000	isu=0.000000
alpha=0.511733	alpha=0.712144
beta=0.819149	beta=0.030930
alpha0=4.499268	alpha0=0.385583

Calibration Conclusions

This is a fairly good result, but in operational applications we would probably seek something nearer to 0.8

Of course, we'd also use a warm-up period and calibration and validation periods

<u>Calibration period</u>: this is the period of time over which the calibration was conducted (in our example, all of 2002)

<u>Validation period</u>: this is an *different* period over which you run the model with the calibrated parameters

Usually, but not always, the validation period will have worse statistics than the calibration period

Calibration Conclusions

This illustrates the reason why we use validation periods: to ensure that the calibrated parameters work for other periods in the time series, not just the period in which the calibration was conducted

Now, let's run EF5 using the calibrated parameters

Copy your calibrated parameters from `cali_dream.puenteportillo.crest.csv` **into your control file in the** `CrestParamSet` **and** `KWParamSet` **blocks**

Comment out the `CalibrateBogota` **task and then uncomment out the** `RunBogota` **task**

control.txt - Notepad

File Edit Format View Help

```
[Task RunBogota]
STYLE=SIMU
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000

[Task CalibrateBogota]
STYLE=cali_dream
MODEL=CREST
ROUTING=KW
BASIN=Bogota
PRECIP=TRMM
PET=FEWSNET
OUTPUT=output\
PARAM_SET=Bogota
ROUTING_PARAM_Set=Bogota
cali_param=Bogota
routing_cali_param=Bogota
TIMESTEP=3h
TIME_BEGIN=200201010300
#TIME_WARMEND=200202010000
TIME_END=200301010000

[Execute]
TASK=RunBogota
#TASK=CalibrateBogota
```

Running the Calibrated Model

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

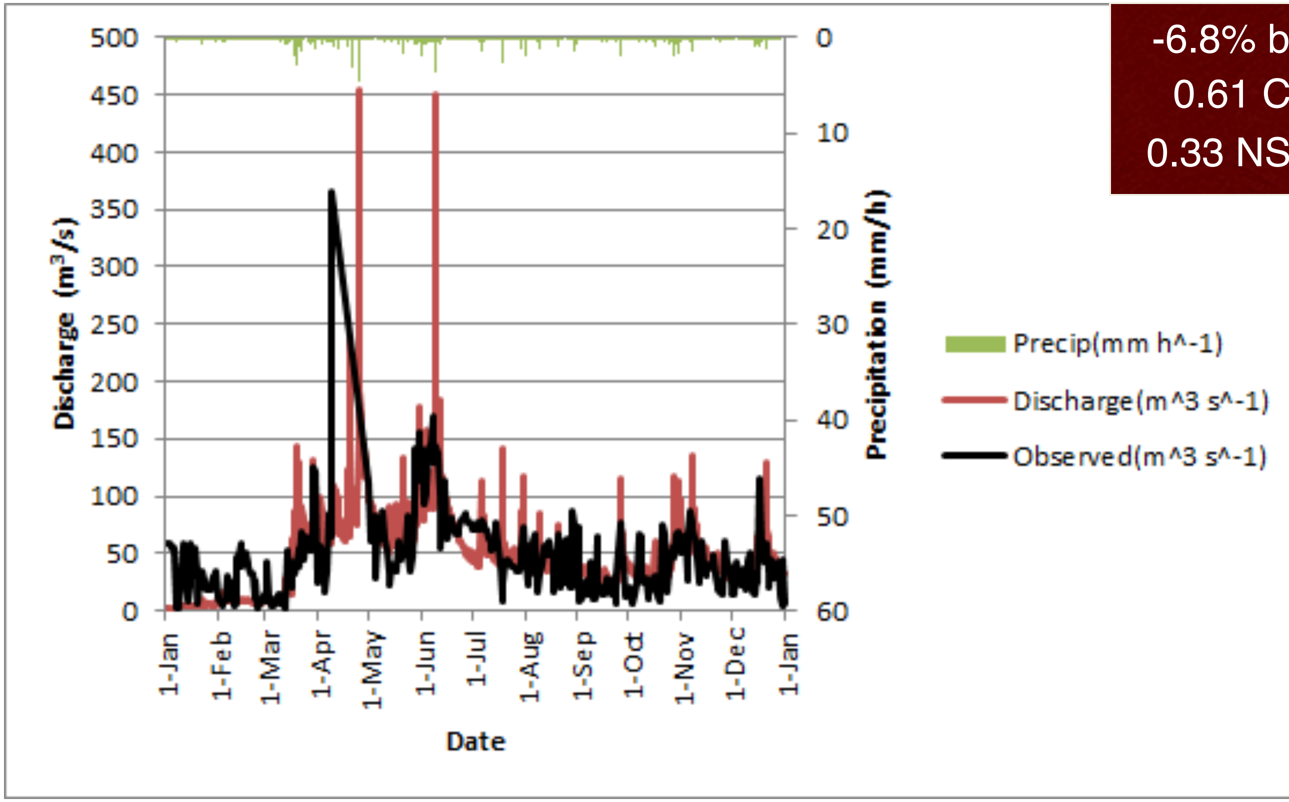
Open the output and start calculating the statistics

- I got a bias of -6.8%, so we started at 67%, then moved to -14%, and now are at -6.8%
- My correlation coefficient was 0.61, so we've come from 0.49, to 0.56, to 0.61
- My NSCE was 0.33, so we have improved from -3.1, to 0.20, to 0.33

Running the Calibrated Model

And here is my final, calibrated hydrograph – much

-6.8% bias
0.61 CC
0.33 NSCE



Note that statistics aren't everything – this is a good result, but we need to work on those big spikes

Coming Up….

The next module is

Using and Interpreting Model Output

You can find it in your `\EF5_training\presentations` directory

Module 2.4 References

EF5 v0.2 Readme, (March 2015).

EF5 Training Doc 2 – Hydrological Model Evaluation, (March 2015).

EF5 Training Doc 4 – EF5 Control File, (March 2015).

EF5 Training Doc 5 – EF5 Parameters, (March 2015).

Guide for Estimating K_{sat} from Soil Properties, Section 618.88 of National Soil Survey Handbook, Natural Resources Conservation Service, United States Department of Agriculture. Available online at http://www.nrcs.usda.gov/wps/portal/nrcs/detail//?cid=nrcs142p2_054224.

Vrugt, J. A., C. J. F. ter Braak, C. Diks, et al. 2009. Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *Int. J. Nonlin. Sci. Num.* 10:273-290.