Get-started manual DDD

Thomas Skaugen, HM/NVE

In order to read csv files the statement now has to be



The term DataFrame has to be inserted after the source file.

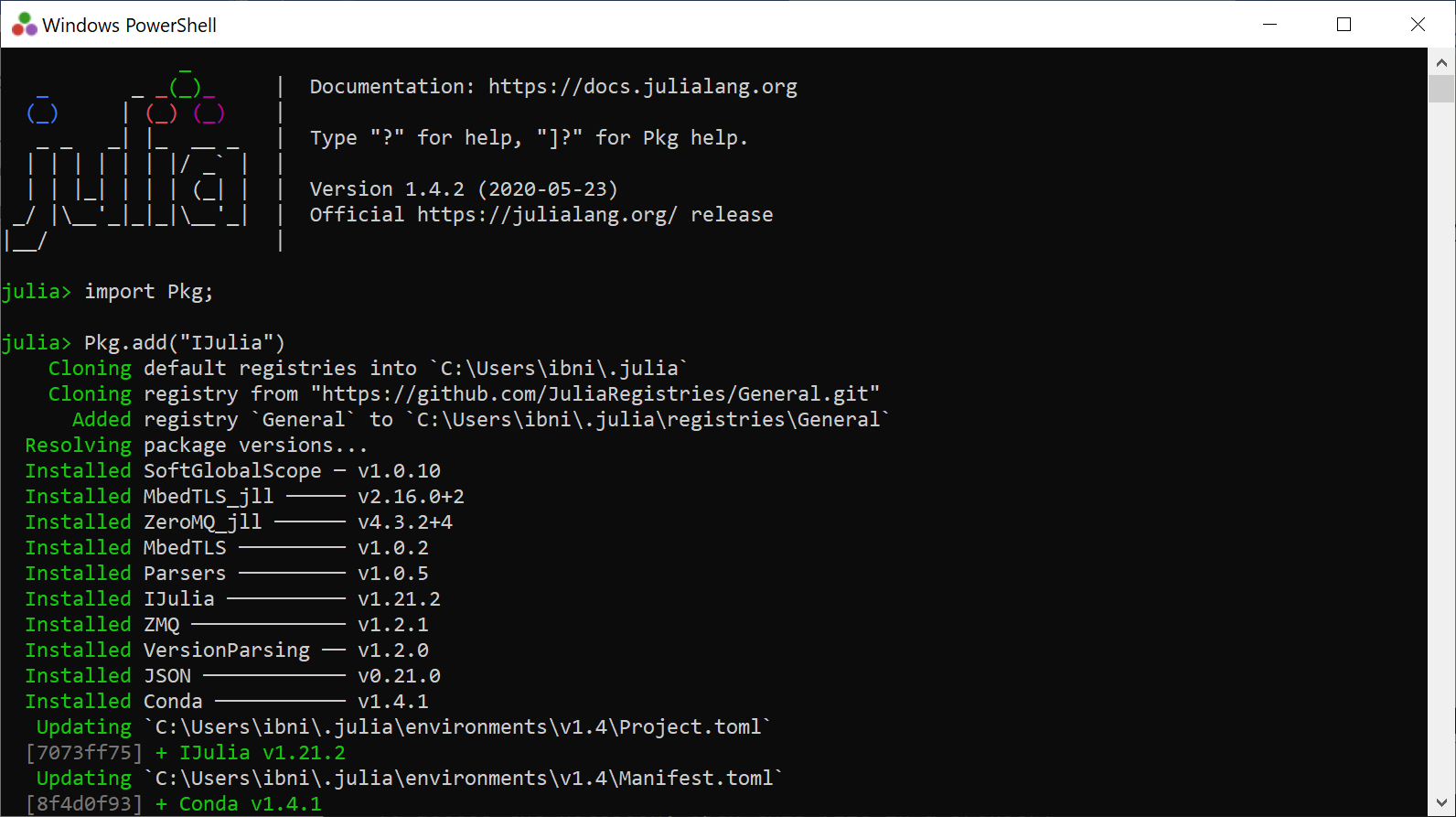
NOTE that not all examples of outputfiles and parameter files in this text are similar to the actual files used. Check with the example files and the code to get the proper structure for these files.

**Downloads**

DDD (Distance Distribution Dynamics) model is a rainfall- runoff model (Skaugen et al, 2020) coded in Julia (download from <https://julialang.org/>) and is edited in Jypyter notebooks (hereafter called «notebook»).

After dowloading Julia and Jypyter, open Julia and write in the Julia window:

“import Pkg; Pkg.add("IJulia") “



This ensures that Julia and Jypyter (in mysterious ways) are connected.

After, download the following packages(while still in the Julia window) needed by DDD. Write:

Pkg.add("CSV")  
Pkg.add("Distributions")  
Pkg.add("LsqFit")  
Pkg.add("Statistics")  
Pkg.add("Dates")  
Pkg.add("DataFrames")

Pkg.add("BlackBoxOptim")  
Pkg.add("Plots")

The package «IJulia» must be installed before you can open a notebook.

**2) Install DDD**



You have to organise subroutines, the main routine and the «run» script appropriately. The above image show how I have organised (which directory) the subroutines.

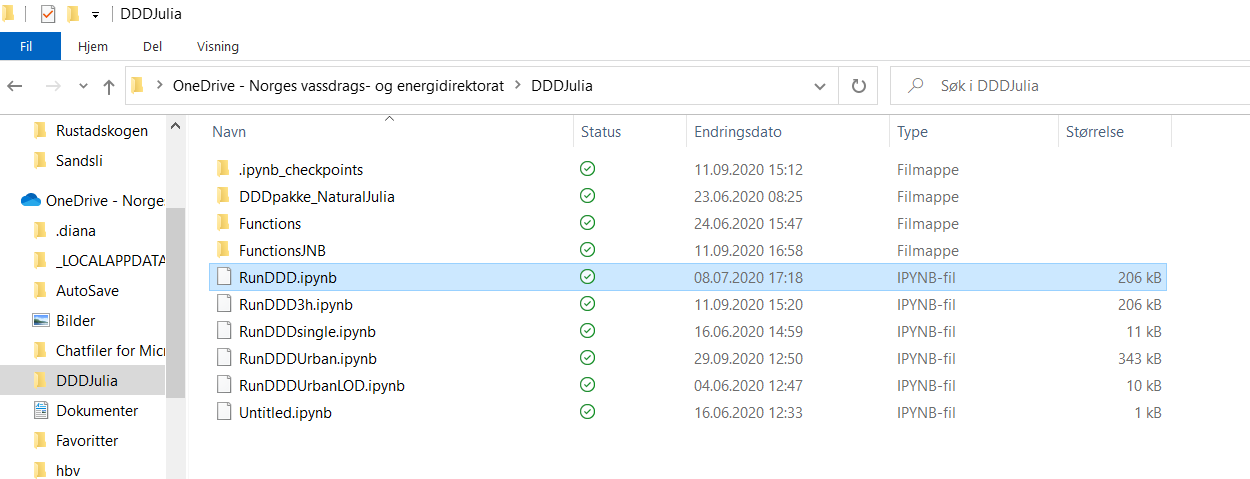
The main routine is located at the same place:

Et bilde som inneholder tekst, skjermbilde, Font, line

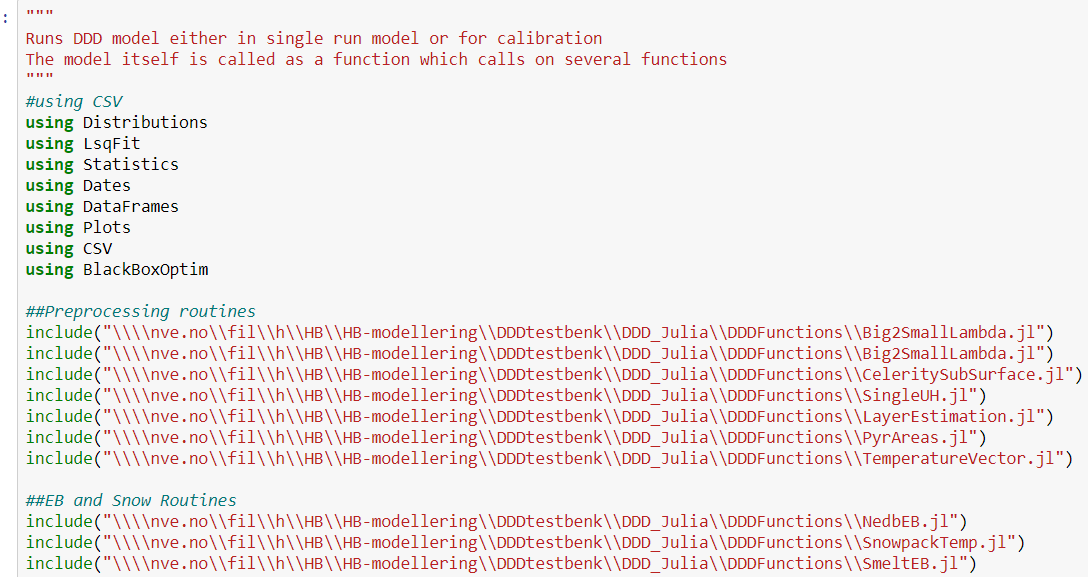
Automatisk generert beskrivelse

This main routine may have changed name due to developments done to the model. This particular version include using an estimate of Mean Annual discharge (MAD) from a regression equation instead of that from the national runoff maps since glacial melt may interfere with the estimate pf forcing due to precipitation alone. In addition, a routine for routing the runoff through lakes is included. NOTE additional parameters are included to what may be described in this text. See example files in GitHub repository.

The «run» script is located on my personal One-drive (experience shows this directory is easily available in Jypyter)



Now open RunDDD.ipynb in notebook og edit all the paths so that they can be found by the program dem. The top of RunDDD may look like this:

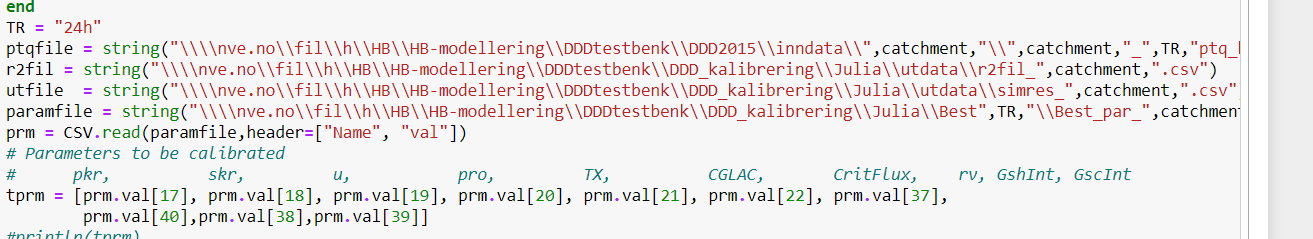


Remember to save (ctrl s)

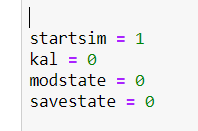
1. **Running a catchment**

Run, for example, the catchmentt (no 31 in the list of catchments, 12.70 Etna) to check that everything works. Remember to edit paths so the the runscript finds the parameterfile (paramfile), input datafile (ptqfile) and where to place the resultfiles (r2fil og utfile).





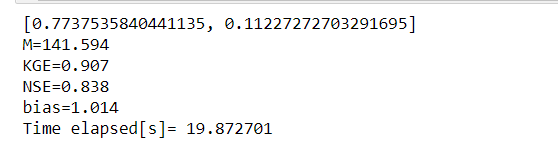
Be sure that «kal=0» when running for the first time. It means that the calibration mode is turned off and you run the catchment with the parameters stated in the parameterfile. If «kal =1» you run a calibration.



You start the script, i.e. run the model by “ctrl Enter”

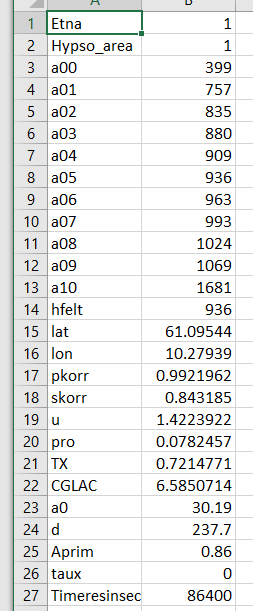
***NB Julia must be started in order to run the model***

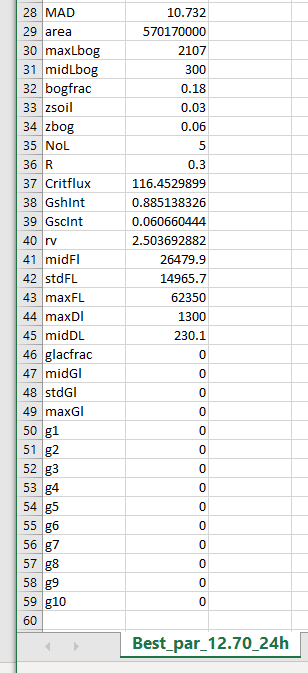
After a few seconds the model is through and the following appear on the screen at the bottom:



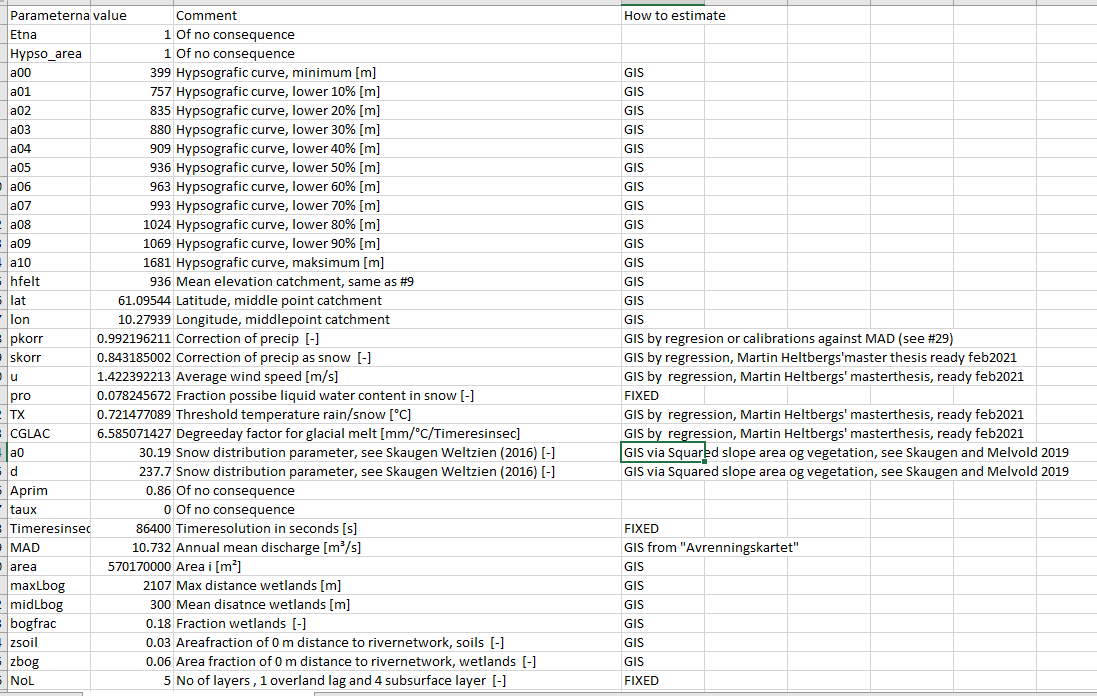
**4) The parameter file**

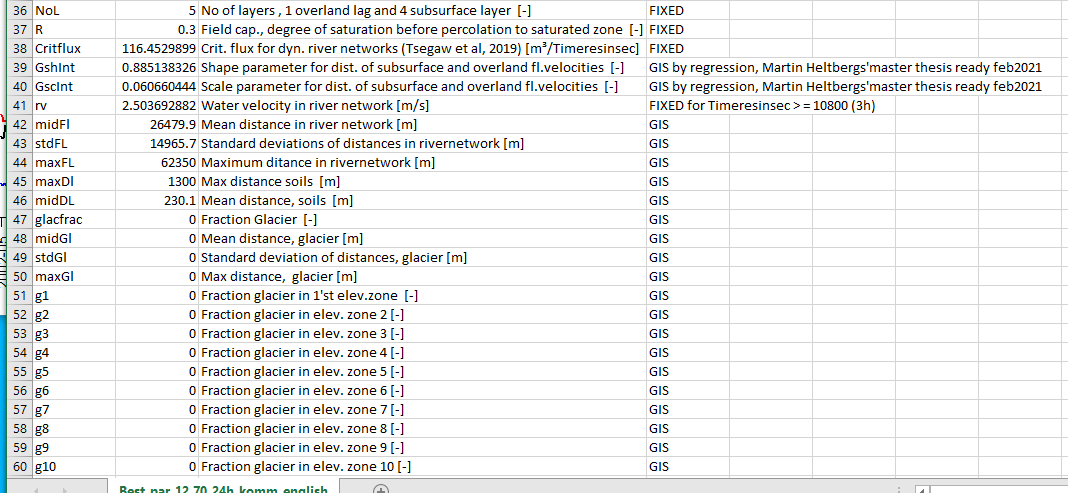
The parameter file is a \*.csv file (which is also the type of the inputfile)





The following images show the same parameters with comments.

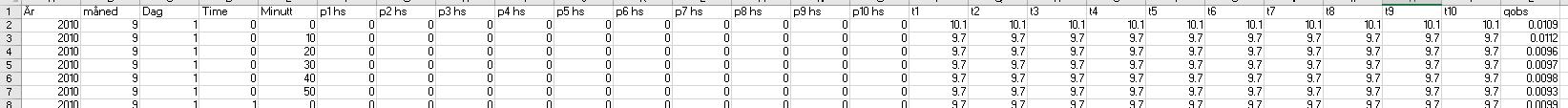




**5) The Input file**

Input to DDD is precipitation and temperature in additions to runoff (necessary only if you want to see the skill score or calibrate, see below) in the temporal resolution of your choice.

The top line is only for illustration purposes, is not to be included in the ptq (p-precip, t-temperature, q- runoff) file. The last column is observed runoff. If runoff is not available, have 0 or some other number, the column needs to have a number. The first 5 columns are the time stamp (for natural catchments there is no column for minutes, i.e only 4 columns for time), the next 10 columns are precipitation for 10 elevation zones, the next 10 columns are, temperature for 10 elevation zones.



**6) Output file**

When «kal=0» there is a result file produced. This is a quite substantial file with 26 columns. The columns are:

1: "yr", year

2: "mnt", month

3: "day", day

4: "hr", hour

5: "p", precipitation [mm]

6: "t", temperature [°C]

7: "qobs", observed discharge [m³/s]

8: "qtot", simulated total discharge [m³/s]

9: "SCA", snow covered area [fraction]

10: "SWE", snow water equivalent [mm]

11: "def1", water in the subsurface [mm]

12: "def2", free capacity in the subsurface (up to M) [mm]

13: "sm", soilwater [mm]

14: "ea", actual evapotranspiration [mm]

15: "outx", water for runoff from soils [mm], incl. overland flow

16: "outbog", water for runoff from wetlands [mm]

17: "outglac", water to runoff glaciers [mm]

18: " m\_r\_ogl", melt and rain from glaciers [mm]

19: "L", water in subsurface including overland flow [mm]

20: " qtot\_mm", simulated runoff in mm [mm]

21: "sm\_Bog", soilmoisture wetlands [mm]

22: "ea\_Bog", actual evapotranspiration wetlands [mm]

23: "L\_Bog", water in subsurface wetlands [mm]

24: "rivstate", water in rivernetwork [mm]

25: "wcs", liquid water in snowpack [mm]

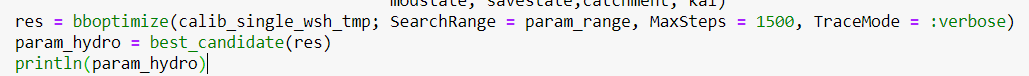
26: "PEvap", potential evapotranspiration [mm]

To read and plot fromt her sult file, use an R-script.

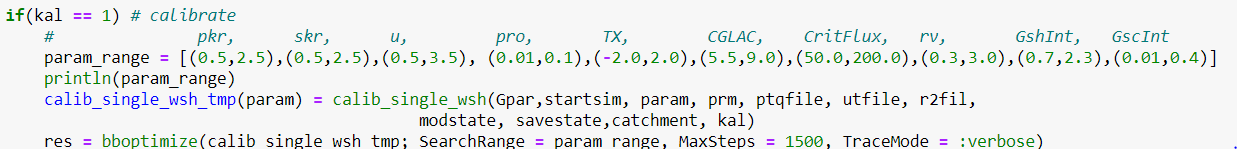
**6) Calibration**

In DDD we have, amongst other parameters, calibrated Gshape and Gscale by optimizing KGE. This is a sensitive affair in that the boundaries of Gshape and Gscale are such that non-pysical combinations of the two, which do not give a proper distribution may occur. In the latest calibration of the operational DDD model Gshape is fixed to 1 (implying an exponential shape), and Gscale can vary.

To calibrate, set «kal=1» in the run script and you choose how many calibration-rounds you want the model to run. If you only calibrate two parameters, the convergence happens quite fast (300 rounds, MaxSteps=300). More parameters demand more rounds, for example 1500 rounds (MaxSteps=1500)



The number of parameters to calibrate is decided by expand or minimize the range of the parameters.



In the example above, every parameter has a range.

**References**

Skaugen T. and C. Onof, 2014. A rainfall runoff model parameterized form GIS and runoff data. *Hydrol. Process*. **28**, 4529-4542,DOI:10.1002/hyp.9968.

Skaugen, T. and Z. Mengistu, 2016. Estimating catchment scale groundwater dynamics from recession analysis- enhanced constraining of hydrological models. *Hydrol. Earth. Syst. Sci.* 20, 4963-4981, doi: 10.5194/hess-20-4963-2016.

Skaugen, T. and Weltzien, I. H., 2016. A model for the spatial distribution of snow water equivalent parameterised from the spatial variability of precipitation, *The Cryosphere*. 10, 1947-1963, doi:10.5194/tc-10\_1947\_2016.

Tsegaw, A.T., Skaugen, T, Alfredsen, K & Muthanna, T.M: [A dynamic river network method for the prediction of floods using a parsimonious rainfall-runoff model](https://iwaponline.com/hr/article/doi/10.2166/nh.2019.003/69546/A-dynamic-river-network-method-for-the-prediction). Hydrology Research 2019; doi.org/10.2166/nh.2019.003, ISSN 1998-9563 (Published online August 26 2019).