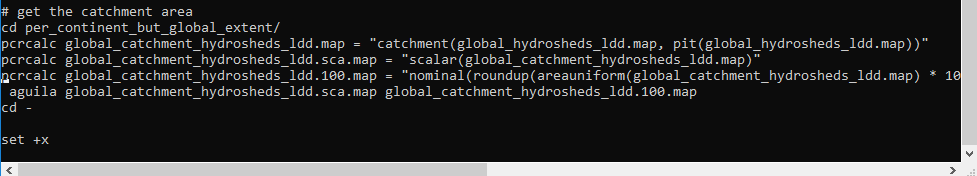
**Workflow PCR-GLOBWB 30-arcsec steady-state GGM**

**Partitioning**

Goal: generate load balanced partitions.

1. Tool: modflow6\trunk\utils\_deltares\metis\partclusshp\partclusshp.f90
2. Input: global\_catchment\_hydrosheds\_ldd\_no\_gl\_aa.idf, created by
   1. global\_catchment\_hydrosheds\_ldd.map from Cartesius  
      /projects/0/dfguu/users/edwinhs/data/global\_hydrosheds\_ldd\_including\_above\_n60/process/version\_20190226   
        
      🡪 global\_catchment\_hydrosheds\_ldd.idf  
      using tool: PCR-GLOBWB-MODFLOW-30arcsec\fortran\pcrlib\vs\pcrlib\x64\Release\pcrlib.exe
   2. Clip for Greenland (GL) and AntArttica using tool  
      PCR-GLOBWB-MODFLOW-30arcsec\fortran\idfclip\vs\x64\Release\idfclip.exe
3. Result: part\_1024\_2.log, part\_1024\_2.idf, sol\_1024\_2.idf  
   (d:\pcr-globwb-1km-model\pcr-globwb-1km\_ldd\)

**Pre-processing PCR-GLOBWB**

Goal: generate .map input rasters for generating the MODFLOW 6 models.

1. Generate 128 clone maps with RCB:
   1. Tool: PCR-GLOBWB-MODFLOW-30arcsec\_clean\fortran\maprcb\src\maprcb.f90
   2. Arg: global\_catchment\_hydrosheds\_ldd.map 128 1
   3. Result: rcb\_???-128.asc; convert to .map files using tool  
      PCR-GLOBWB-MODFLOW-30arcsec\_clean\scripts\gdal\_asc2map.bat
2. Create general .ini file:  
   PCR-GLOBWB-MODFLOW-30arcsec\_clean\config\rcb\template\_2.ini

Run Python script @Cartesius:  
PCR-GLOBWB-MODFLOW-30arcsec\_clean\scripts\clonemaprcb.py  
with arguments with shell script:  
  
#!/bin/bash

script=../PCR-GLOBWB-MODFLOW-30arcsec/scripts/clonemaprcb\_2.py

python ${script} -r -a -np 128 -fs 10 40 75 1000 -th\_a 0 2 5 10 -tm\_a 30 0 0 0 -q\_a short normal normal normal -mapdir ../clonemap

1. Zip the result map data files @Cartesius using tool PCR-GLOBWB-MODFLOW-30arcsec\_clean\scripts\rcb\_zip.py
2. Merge the .map file tot he global domain using script  
   PCR-GLOBWB-MODFLOW-30arcsec\_clean\scripts\mergemap2idf.bat

**Model generation**

Goal: Generate dependent and independent MODFLOW 6 models

1. Tool: PCR-GLOBWB-MODFLOW-30arcsec\fortran\mf6ggm\vs\x64\Release\mf6ggm.exe
2. Input (mf6ggm.inp):
   1. part\_1024\_2.log
   2. TOP .\input\_data\top\_uppermost\_layer.idf
   3. BOT\_L1 .\input\_data\bottom\_uppermost\_layer.idf
   4. BOT\_L2 .\input\_data\bottom\_lowermost\_layer.idf
   5. K\_L1 .\input\_data\horizontal\_conductivity\_uppermost\_layer.idf
   6. K\_L2 .\input\_data\horizontal\_conductivity\_lowermost\_layer.idf
   7. K33\_L1 .\input\_data\vertical\_conductivity\_uppermost\_layer.idf
   8. K33\_L2 .\input\_data\vertical\_conductivity\_lowermost\_layer.idf
   9. STRT\_L1 .\input\_data\initial\_head\_uppermost\_layer.idf
   10. STRT\_L2 .\input\_data\initial\_head\_lowermost\_layer.idf
   11. DRN\_ELEV\_L1 .\input\_data\drain\_elevation\_uppermost\_layer.idf
   12. DRN\_ELEV\_L2 .\input\_data\drain\_elevation\_lowermost\_layer.idf
   13. DRN\_COND .\input\_data\drain\_conductance.idf
   14. RIV\_STAGE\_L1 .\input\_data\surface\_water\_elevation.idf
   15. RIV\_RBOT\_L1 .\input\_data\surface\_water\_bed\_elevation\_used.idf
   16. RIV\_COND .\input\_data\bed\_conductance\_used.idf
   17. RECHARGE .\input\_data\net\_RCH.idf
   18. PARTITIONS part\_1024\_2.idf
   19. SOLUTIONS sol\_1024\_2.idf
3. Result: MODFLOW 6 models

**Generating and running the models**

Tool: PCR-GLOBWB-MODFLOW-30arcsec\_clean\scripts\run\_models.py

Three steps:

1. *Option -pre*:  
   Model generation. Call mf6ggm.exe (see above) with arguments:  
   <sol number> <solution folder> mf6ggm.inp
2. *Option -run\_ser*:  
   Besides external Dirichlet boundary (= 0), also impose internal boundary for improving starting head  
   🡪 Run separate models by looping over all separate s<##>,m<#####>.mfsim files  
   🡪 This overwrites m<#####>.int.ext in the s<##>\results folder
3. *Option run\_par*:  
   Only impose external Dirichlet boundary condition

@Windows server (full model exterior boundary only):

1. s1-s38 Independent serial model, using   
   PCR-GLOBWB-MODFLOW-30arcsec\_clean\scripts\run\_models\_indep.bat
2. s39-s48 dependent small parallel models up to 8 cores, using  
   PCR-GLOBWB-MODFLOW-30arcsec\_clean\scripts\run\_models\_dep.bat
3. Full model internal & external boundary

@Cartesius (s49, s50 and s51):

1. Model generation by calling mf6ggm directly. Scripts:  
   init\_model\_s49, init\_model\_s50 , init\_model\_s51vi  
   (/home/jarno2/ggm-1km/jobs)

Script order:

run\_serial\_model\_s51  
s51\_n51m612\_2

**Post-processing results**