HGS-PDAF v1.0 (III)

Qi Tang
University of Neuchâtel

28.11.2023

unine
Université de Neuchâtel
Faculté des sciences

WATERAGRI

HGS-PDAF: preparation with an example model

Qi Tang[1,2], Hugo Delottier[1], Wolfgang Kurtz[3], Lars Nerger[4], Oliver Schilling[1,2,5], Philip Brunner[1]

[1] Centre for Hydrogeology and Geothermics (CHYN), University of Neuchâtel, Switzerland

[2] Hydrogeology, Department of Environmental Sciences, University of Basel, Switzerland

[3] Agrometeorology, Branch Office Weihenstephan, German Meteorological, Germany

[4] Helmholtz Centre for Polar and Marine Research, Alfred Wegener Institute, Germany

[5] Eawag, Swiss Federal Institute of Aquatic Science and Technology, Switzerland

Where to find the example models

- If you use git:
  - Github repository for users: https://github.com/qiqi1023t/HGS-PDAF_v1.0_GMD
  - Use git clone or svn checkout to download the repository to your local machine
  - The example folder is included together with the source code
- Or download the current version on Zenodo:
  - https://doi.org/10.5281/zenodo.10000887
  - Filename: examples.zip
- Follow the Readme file in the source code to get the necessary input files for the model

What are we going to prepare?

- Open the runscript:

```
# Define run directory
export BASE_DIR=   # root path to the hgs-pdaf directory
export HGS_DIR=   # root parth to HGS binary
export MODEL_DIR=  # root path to the HGS model directory
export WORK_DIR=  # root path to the output directory
export ENS_DIR=  # root to the output ensemble directory
```

```
export BASE_DIR=/p/home/jusers/tang1/jusuf/HGS-PDAF_V1.0/hgs-pdaf
export HGS_DIR=/p/home/jusers/tang1/jusuf/HydroGeoSphere-2529-
Linux/HydroGeoSphere-2529-Linux/
export MODEL_DIR=/p/project/icei-prace-2023-0004/tang1/input_HGS-
PDAF/hgsmodel/synthetic_hete/
export WORK_DIR=/p/scratch/icei-prace-2023-0004/tang1/output/hgs-pdaf
export ENS_DIR=${WORK_DIR}/n2_hete/
```

Step 1: prepare the HGS model

- Adapt the transient HGS model to multiple 1-step models
- This has already been done for our example case
- If the user starts with his/her own model, this needs to be done before the DA run!
- The following slides show how to do this

Adapt the HGS model to run hgs-pdaf

- Main idea:
  - The transient model is split into multiple short-period model. The length of this short period is equal to the interval between two assimilation steps;
  - After each time step, the initial conditions for the next time step will be replaced by the out put from the current time step;
  - Boundary conditions at each time step is replaced with the real value.

Example of bash scripts to prepare the HGS model

```
#!/bin/bash

python EXECPROC/Preproc.py
python EXECPROC/Preproc_pdaf.py

cp HGS/Grokfiles/wells/wells_flow_1.inc HGS/Grokfiles/wells_flow.inc
python EXECPROC/Spinup.py


for i in $(seq 1 1 95)
do
cp HGS/Grokfiles/wells/wells_flow_$i.inc HGS/Grokfiles/wells_flow.inc

python EXECPROC/Execproc.py
python EXECPROC/obshgs2obspest_Sequential.py

preplot HGS/Flowo.pm

cp HGS/Flowo.head_pm.0001 HGS/IC/Ini_pm_SS
cp HGS/Flowo.head_olf.0001 HGS/IC/Ini_olf_SS

done
#SLEEP 3
#)
```

WATERAGRI

Step 2: prepare the observation file

- Generate the head observation
- The observation file for this test case is already prepared and included in the model folder.
- If the user starts with his/her own model, this needs to be done before the DA run!
- The following slides show how to do this

Find the observation location file

Prepare the file '*ObservationPoints_selection.dat*' in the preprocessing folder. The location (x,y,z coordinates) is defined in this file.

```
55,50,85
255,50,85
95,150,85
35,200,85
255,250,85
55,300,85
235,350,85
55,450,85
```

Find the observation value file

The observation values are stored in separated files. In this example we have 8 obervation points, i.e. 8 files.

Observation 1:

```
365,94.18771457,0.05
366,94.2420817,0.05
367,94.13884553,0.05
368,94.14725778,0.05
369,94.09936834,0.05
370,94.09046822,0.05
371,94.06449754,0.05
372,94.09066345,0.05
373,94.1509718,0.05
374,94.17967425,0.05
375,94.11391162,0.05
376,94.16649582,0.05
377,94.14672236,0.05
378,94.14981592,0.05
```

Example of the generated observation file

Copy the HGS generated coordinate file '*Flowo.coordinates_pm*' to the preprocessing folder
Run *gen_obs_head*
You will get the observation file *obs_HEAD.nc*

```
netcdf obs_HEAD {
dimensions:
        n_obs = 8 ;
        time = 95 ;
variables:
        float x(n_obs) ;
        float y(n_obs) ;
        float z(n_obs) ;
        int obs_id(n_obs) ;
        float time(time, n_obs) ;
        float Head(time, n_obs) ;
        float std(time, n_obs) ;

// global attributes:
                :title = "HEAD observations" ;
                :_FillValue = 1.e+07f ;
```

Step 3: Modify the configuration files for the parameters

- Modify the two *namelist* files to specify the parameters for HGS and PDAF

Step 3: modify the namelist files for HGS

- Modify *namelist.hgs*

```
! Namelist for HGS model configuration
&hgs

! Settings for model initialisation
prefix = 'Flow'
trace_name = ''
insuffix = '0001' ! last state output No., e.g., '.0024'
outsuffix = '0001' ! new ini state No., e.g., '.0000'
isolf = .true. ! true if overland flow, false if only pm flow
isconc = .false. ! true with transport, false without
hgs_version = 2 ! 1 for old HGS version, 2 for HGS 2013+

/
```

Step 3: modify the namelist files for PDAF

- Modify *namelist.pdaf*

```
! Namelist for PDAF configuration
&pdaf
! General control
str_daspec='da'
filtertype=2
printconfig=.true.
locweight=3
loctype=0
forget=1.0
varscale=1.0
type_trans=0
type_forget=0
use_global_obs = .true.
istep=<pdafstep>
```

Prefix for the data assimilation output file

Assimilation method

For localised filter, currently not available

For inflation

Step 3: modify the namelist files for PDAF

```
! Settings for initialization
path_init=''
file_init=''
read_inistate=.false.
file_inistate='state_ini_'

! Output control
write_da = .true.
write_ens = .true.
/

! Namelist for parallel configuration for ensemble DA
&pdaf_parallel
dim_ens=<ens_size>
/
```

Write the DA output file
Write the result for each ensemble member

Step 3: modify the namelist files for PDAF

```
! Settings for assimilating HEAD observations
&pdaf_hgs
state_type = 6
assim_o_head=.true.
rms_obs_head=0.05
lradius_head=3.0e5
sradius_head=3.0e5
head_fixed_rmse=.true.
head_exclude_diff=1.6
path_obs_head='/p/project/icei-prace-2023-0004
file_head_prefix='obs_HEAD'
file_head_suffix ='.nc'
```

Which variables will be included into the state vector

Observation error

Full name of the observation file

Step 3: modify the namelist files for PDAF

```
! Settings for assimilating SATURATION observations
assim_o_sat=.false.
rms_obs_sat=0.01e8
lradius_sat=3.0e5
sradius_sat=3.0e5
sat_fixed_rmse=.true.
sat_exclude_diff=1.6
path_obs_sat='/p/project/icei-prace-2023-0004/tang1/i
file_sat_prefix='obs_SAT'
file_sat_suffix ='.nc'

ResultPath=<path>
/
```

Step 4: complete the run script

- Modify the run script
- The following slides show an example of the run script used for the slurm system

Step 4: Complete the run script

```
#!/bin/bash -x
#SBATCH --account=
#SBATCH --nodes=
#SBATCH --ntasks=
#SBATCH --ntasks-per-node=
#SBATCH --output=
#SBATCH --error=
#SBATCH --time=
#SBATCH --partition=batch
#SBATCH --mail-user=
#SBATCH --mail-type=ALL
```

Your account

Number of nodes

Number of tasks

Number of tasks per node

Name of the job output file

Name of the job error output file

Required run time

You email address

Step 4: Complete the run script

```
# *** start of job script ***
ulimit -s unlimited
set -vx
export NOPP=1

echo $NCPUS
# system dependent command lines
#module load Stages/2019a
module load Intel
module load ParaStationMPI
module load imkl netCDF netCDF-Fortran
export NETCDF_Fortran_INCLUDE_DIRECTORIES=${EBROOTNETCDFMINFORTRAN}/include/
export NETCDF_C_INCLUDE_DIRECTORIES=${EBROOTNETCDF}/include/
export NETCDF_CXX_INCLUDE_DIRECTORIES=${EBROOTNETCDFMINCPLUSPLUS4}/include/

module load Python
module load SciPy-Stack

# end

export OMP_WAIT_POLICY=PASSIVE
export CRAY_OMP_CHECK_AFFINITY=TRUE
export OMP_NUM_THREADS=1
```

You may need to change these lines according to your system configuration

Step 4: Complete the run script

```
# Ensmeble size
export NENS=2

# Total time steps
export TSTEPS=2

# Define run directory
export BASE_DIR=   # root path to the hgs-pdaf directory
export HGS_DIR=   # root parth to HGS binary
export MODEL_DIR=  # root path to the HGS model directory
export WORK_DIR=  # root path to the output directory
export ENS_DIR=  # root to the output ensemble directory

# Define if serial model run or DA run
# A serial run should be done before the ensemble run
export firstrun=T
```

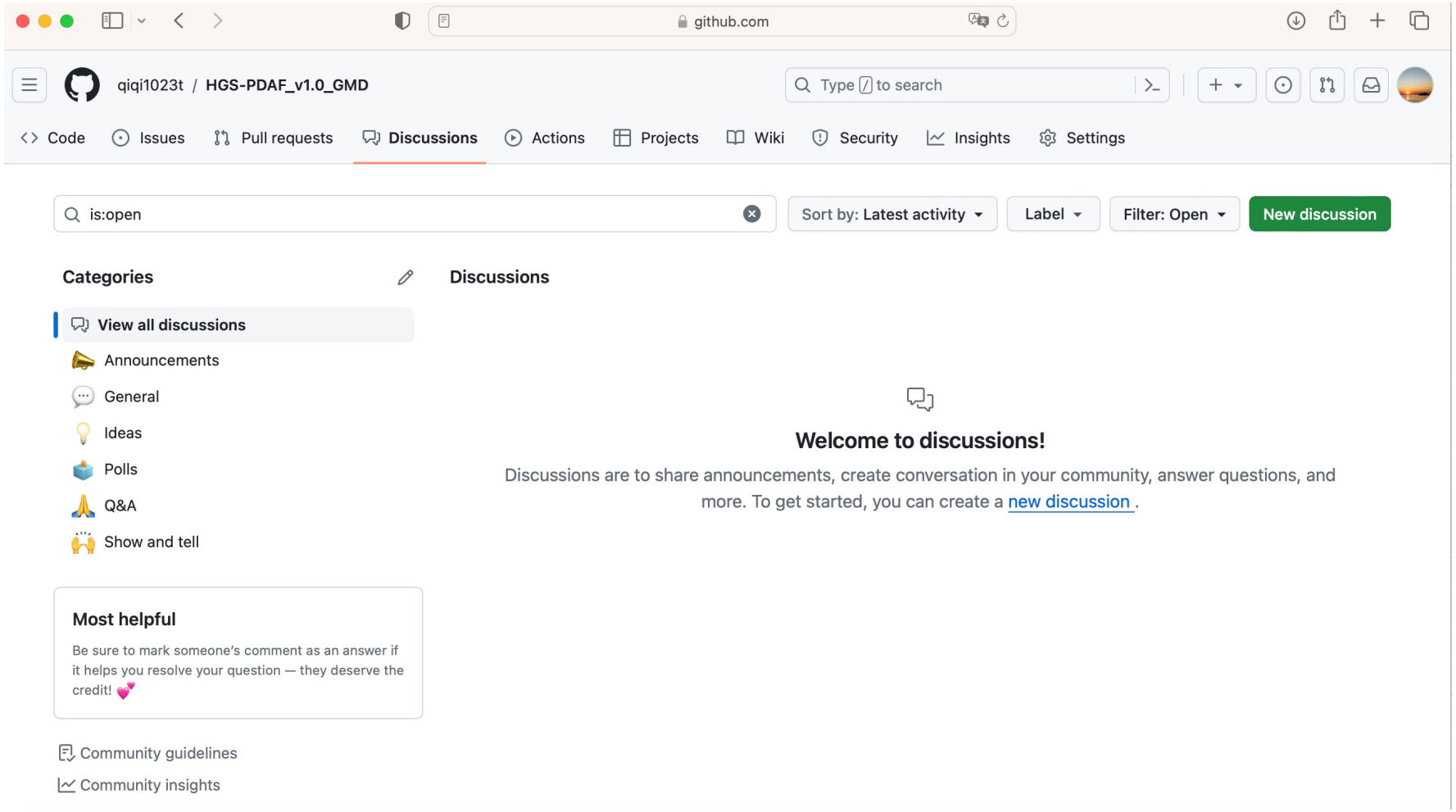
If this is set to F(False), spin-up run will be skipped.
Always keep this as T if parameters are updated.

Step 5: Run the experiment

- Submit the run script by typing:

```
sbatch name_of_the_runscript.sh
```

- You will find the output files in the `$WORK_DIR` directory



Discussion on GitHub

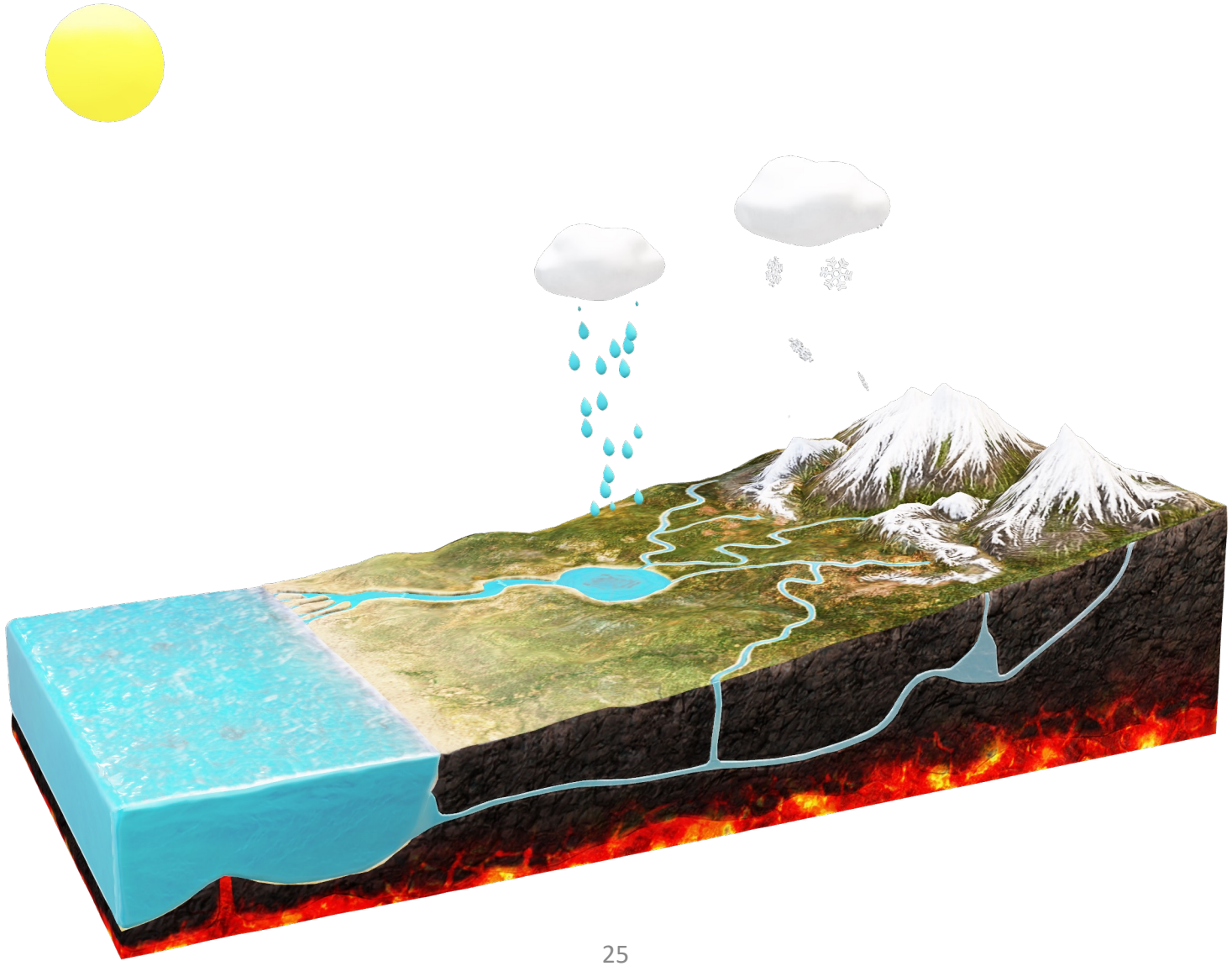
If you want to discuss more with other users/developers:

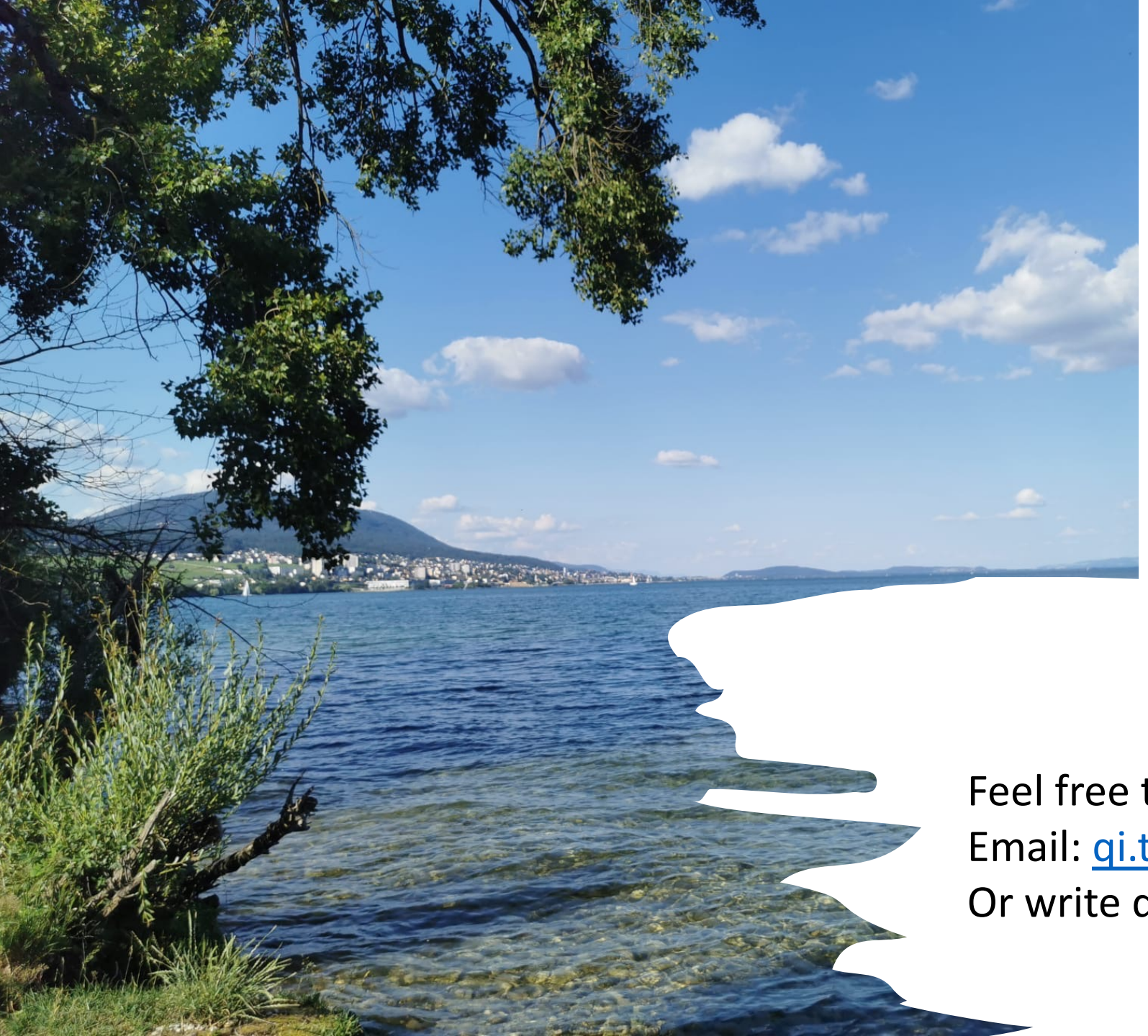
For advanced user:

- If you need
  - Assimilate a new observation type
  - Adding a new variable into the state vector
  - Using different types of filters
  - Use localisation
  - Use heterogeneous observation errors
  - Other requirements regarding to the specific assimilation approach

Please let me know and I'm more than happy to help!
Contact: qi.tang@unine.ch

Next step: Start your own data assimilation experiments





Enjoy it!

Feel free to contact us if you need any help:

Email: qi.tang@unine.ch

Or write directly on GitHub!