



HGS-PDAF v1.0 (II)

Qi Tang
University of Neuchâtel

28.11.2023

HGS-PDAF: modules, subroutines and functionality

Qi Tang^{1,2}, Hugo Delottier¹, Wolfgang Kurtz³, Lars Nerger⁴, Oliver Schilling^{1,2,5}, Philip Brunner¹

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

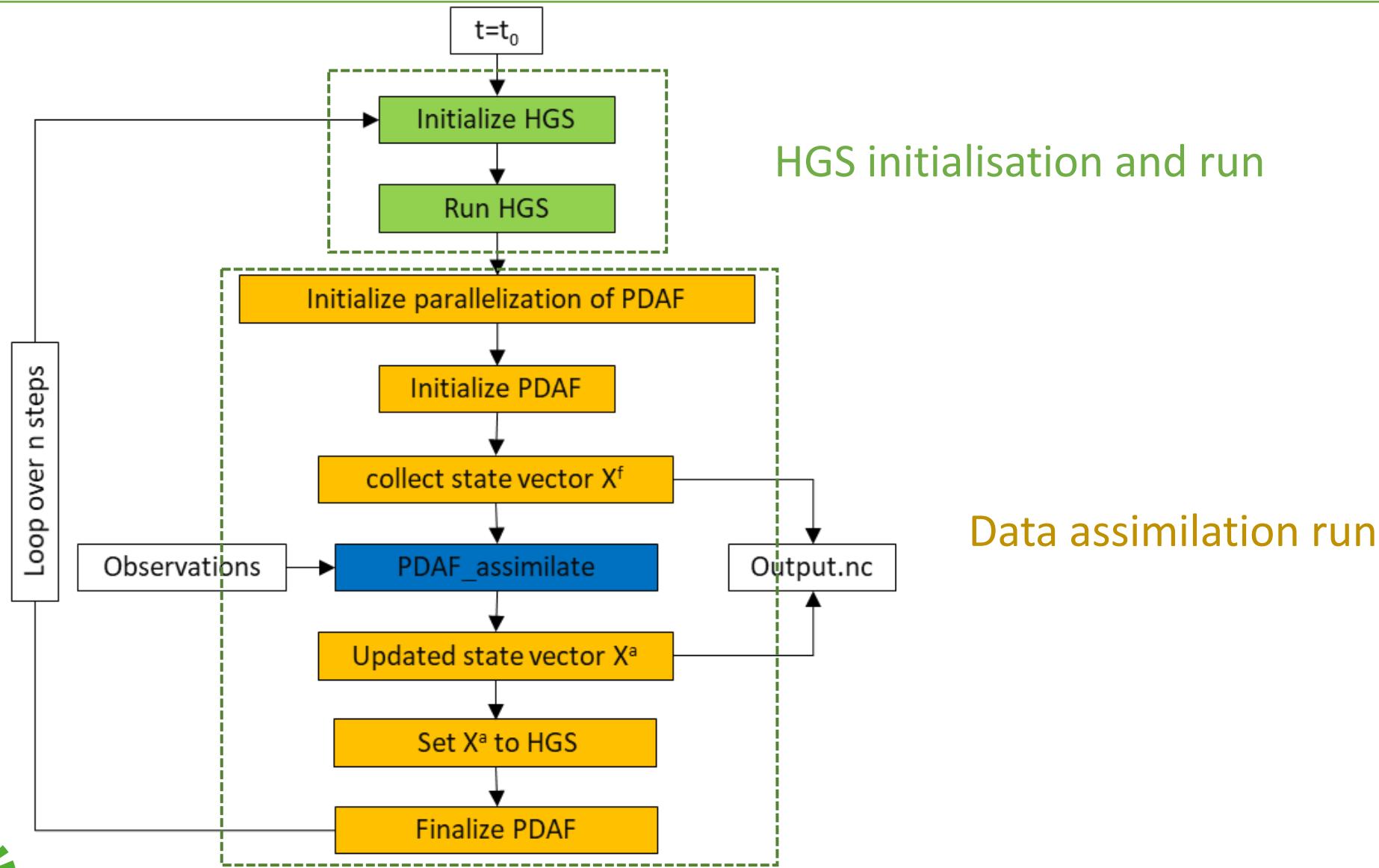
² Hydrogeology, Department of Environmental Sciences, University of Basel, Switzerland

³ Agrometeorology, Branch Office Weihenstephan, German Meteorological, Germany

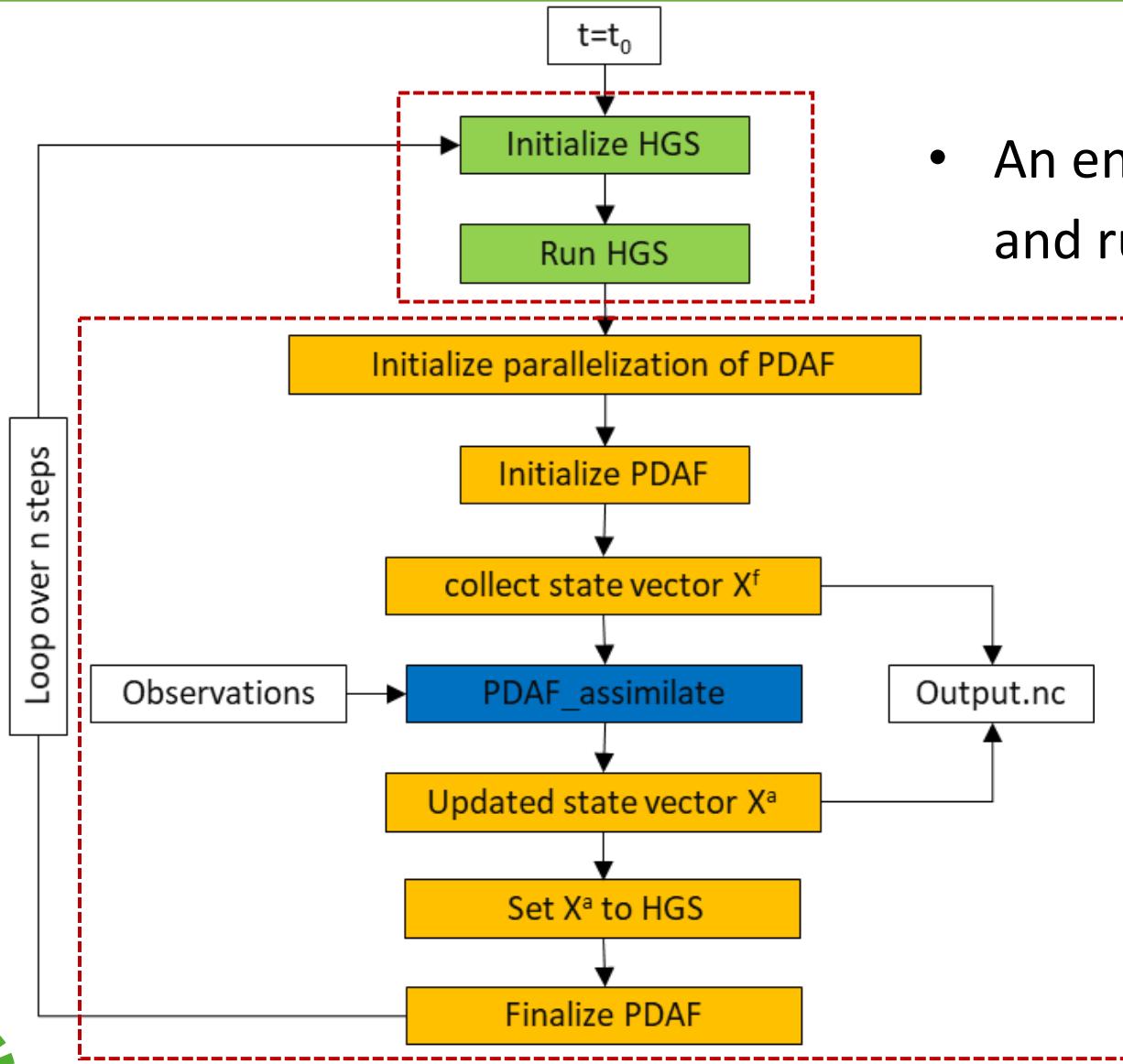
⁴ Helmholtz Centre for Polar and Marine Research, Alfred Wegener Institute, Germany

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

Workflow



Workflow



- An ensemble of HGS models is initialized and run in parallel
- Initialization of data assimilation
- Colleting model simulation from HGS output
- Observation is read and operated
- Data assimilation is carried out
- Model simulation is updated
- Finalisation

Where to find executables and tools

- 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
 - Keep updated with the new version
- Or download the current version on Zenodo:
 - <https://doi.org/10.5281/zenodo.10000887>
 - Filename: HGS-PDAF_v1.0.zip



Github

The screenshot shows a GitHub repository page for 'HGS-PDAF_v1.0_GMD'. The repository is public and has 1 branch and 0 tags. The most recent commit was made by Qi Tang 31 minutes ago, updating the README file. The repository contains files for examples, hgs-pdaf, hgsilib, namelists, postprocessing, preprocessing, run_scripts, LICENSE, and README.md. The 'About' section indicates no description, website, or topics provided. There are 0 stars, 1 watching, and 0 forks.

HGS-PDAF_v1.0_GMD Public

Qi Tang update the readme file 9737790 31 minutes ago 8 commits

File	Commit Message	Time
examples/3-D_synthetic_model	Add examples	1 hour ago
hgs-pdaf/offline_omi	hgs-pdaf model bindings	last week
hgsilib	hgs-pdaf model bindings	last week
namelists	hgs-pdaf model bindings	last week
postprocessing	hgs-pdaf model bindings	last week
preprocessing	hgs-pdaf model bindings	last week
run_scripts	hgs-pdaf model bindings	last week
LICENSE	Initial commit	last week
README.md	update the readme file	31 minutes ago

About

No description, website, or topics provided.

Readme
GPL-3.0 license
Activity
0 stars
1 watching
0 forks

Releases

No releases published
[Create a new release](#)

Zenodo

The screenshot shows a Zenodo repository page for "HGS-PDAF (version 1.0): The PDAF model bindings and examples for HydroGeoSphere".

Header: zenodo.org

Search bar: Search records...

User menu: Communities, My dashboard, +, User icon: qi.tang10...

Published: October 13, 2023 | Version v1

Actions: Journal article, Open, Edit, New version, Share

Metrics: 7 Views, 0 Downloads

Affiliations: Tang, Qi^{1,2} (ORCID), Delottier, Hugo¹ (ORCID), Kurtz, Wolfgang³ (ORCID), Nerger, Lars⁴ (ORCID), Schilling, Oliver S.^{2,5} (ORCID), Brunner, Philip¹ (ORCID). Hide affiliations.

Contributors:

1. ROR University of Neuchâtel
2. ROR University of Basel
3. ROR German Meteorological Service
4. ROR Alfred Wegener Institute for Polar and Marine Research
5. ROR Swiss Federal Institute of Aquatic Science and Technology

Description: This repository contains the model source code, input/output scripts and data used to produce the results of the paper 'HGS-PDAF (version 1.0): A coupled data assimilation framework for an integrated surface and subsurface hydrological model' in GMD. The following two parts are included:

- The source code of HGS-PDAF_v1.0, together with the post-processing Python scripts to analyse and visualise the output files;
- The examples illustrating HGS-PDAF, including all input files and the output data for the open-loop run and the data assimilation experiments. All output data are stored in separate files named by their scenario names.

Versions:

Version v1	Oct 13, 2023
10.5281/zenodo.10000887	

Cite all versions? You can cite all versions by using the DOI [10.5281/zenodo.10000886](https://doi.org/10.5281/zenodo.10000886). This DOI represents all versions, and will always resolve to the latest one. [Read more](#).

Code structure

 docs	<ul style="list-style-type: none">• Documentation
 examples/3-D_synthetic_model	<ul style="list-style-type: none">• Examples to run <i>hgs-pdaf</i>
 hgs-pdaf/offline_omi	<ul style="list-style-type: none">• Source code of <i>hgs-pdaf</i>
 hgsiolib	<ul style="list-style-type: none">• Subroutines to read and write HGS original output, written in Fortran
 namelists	<ul style="list-style-type: none">• Configuration files where parameters are defined
 postprocessing	<ul style="list-style-type: none">• Postprocessing: Python script to visualize DA output
 preprocessing	<ul style="list-style-type: none">• Preprocessing: routines to prepare the observation written in Fortran
 run_scripts	<ul style="list-style-type: none">• Bash script to manage the DA simulation runs
 LICENSE	<ul style="list-style-type: none">• LICENSE
 README.md	<ul style="list-style-type: none">• Readme file

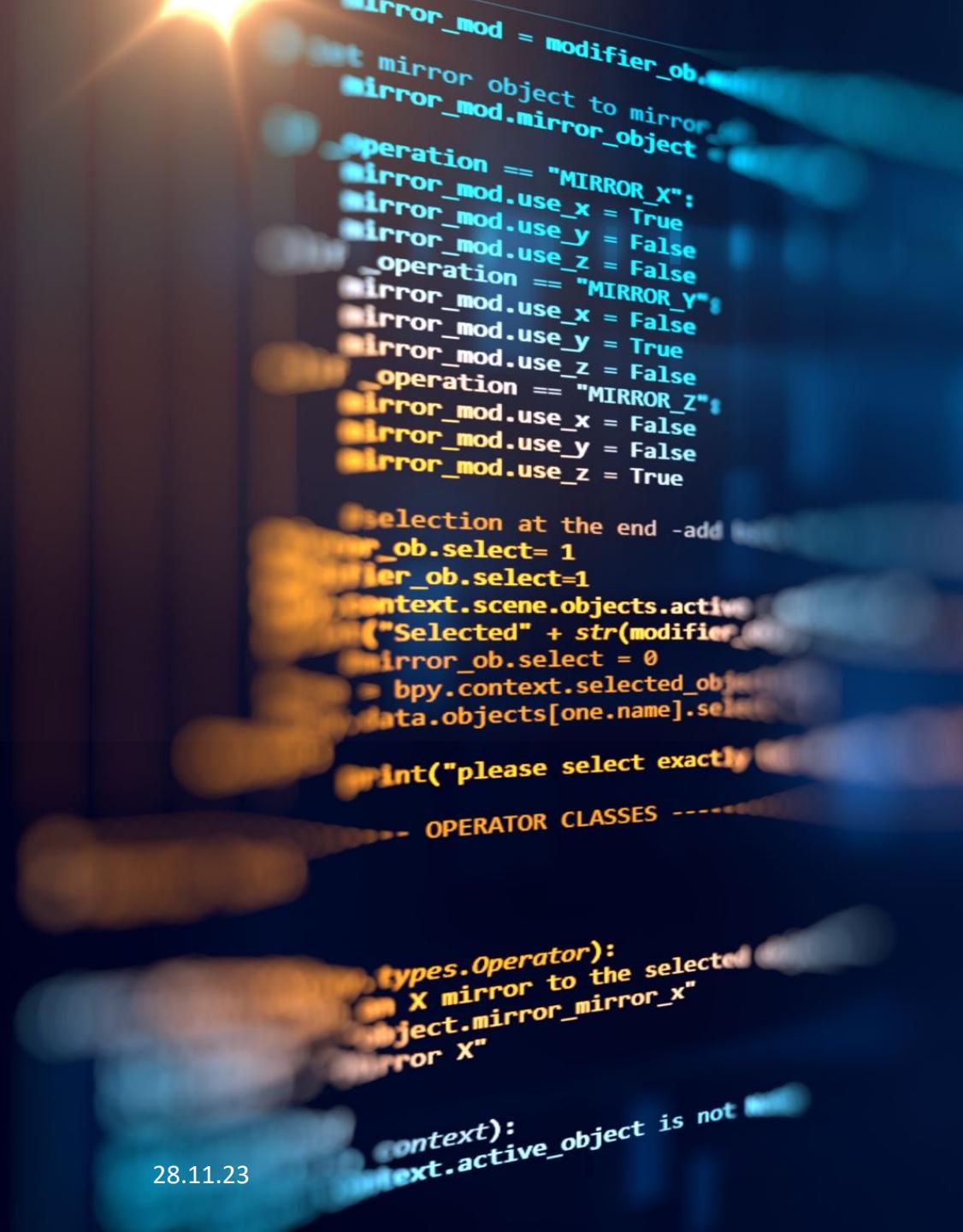


Code structure

Enter the main directory:

```
[[tang1@jsf102 HGS-PDAF_v1.0_GMD]$ ll
total 57
-rw-r--r-- 1 tang1 jusers 35149 Nov 20 21:52 LICENSE
-rw-r--r-- 1 tang1 jusers 3423 Nov 28 12:38 README.md
drwxr-xr-x 2 tang1 jusers 4096 Nov 28 13:30 docs
drwxr-xr-x 3 tang1 jusers 4096 Nov 28 11:41 examples
drwxr-xr-x 3 tang1 jusers 4096 Nov 20 21:52 hgs-pdaf
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 hgsiolib
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 namelists
drwxr-xr-x 4 tang1 jusers 4096 Nov 20 21:52 postprocessing
drwxr-xr-x 4 tang1 jusers 4096 Nov 20 21:52 preprocessing
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 run_scripts
```



A blurred background image of a computer screen displaying Python code for a Blender operator. The code is related to the 'MIRROR' modifier and includes logic for selecting objects based on their type and active status.

```
mirror_mod = modifier_obj
# set mirror object to mirror
mirror_mod.mirror_object = None
operation = "MIRROR_X"
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation == "MIRROR_Y"
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation == "MIRROR_Z"
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True

if selection at the end -add
    mirror_ob.select= 1
    mirror_ob.select=1
    context.scene.objects.active = mirror_ob
    ("Selected" + str(modifier))
    mirror_ob.select = 0
    bpy.context.selected_objects = []
    data.objects[one.name].select = 1
    print("please select exactly one object")
else:
    print("please select exactly one object")

-- OPERATOR CLASSES ---

@types.Operator:
    X mirror to the selected object.mirror_mirror_x"
    "mirror X"
    "mirror Y"
    "mirror Z"

@context:
    "context.active_object is not None"
```

Compiling the code

Follow the Readme file to compile the source code to create **three executables/libraries** in different folders

HGS I/O subroutines

```
[[tang1@jsf102 hgsiolib]$ ll
total 403
-rw-r--r-- 1 tang1 jusers  19472 Jun 22 16:47 hgs_fun.f90
-rw-r--r-- 1 tang1 jusers 161136 Aug  6 14:44 hgs_fun.o
-rw-r--r-- 1 tang1 jusers   2955 Aug  6 14:44 hgs_mod.F90
-rw-r--r-- 1 tang1 jusers  11032 Aug  6 14:44 hgs_mod.o
-rw-r--r-- 1 tang1 jusers    1049 Apr 13 2023 hgs_read.h
-rw-r--r-- 1 tang1 jusers    1364 Apr 13 2023 hgs_read_test.h
-rw-r--r-- 1 tang1 jusers   9395 Aug  6 14:44 hgsdat.mod
-rw-r--r-- 1 tang1 jusers 174236 Aug 25 10:32 libhgs.a
-rw-r--r-- 1 tang1 jusers     700 Aug  6 14:44 machine_constants.mod
-rw-r--r-- 1 tang1 jusers    539 Apr 13 2023 makefile
```

Function: Read/Write original HGS binary file.



PDAF source code

- PDAF source code is included in the folder
- You can also download the latest (full) version of PDAF from <https://pdaf.awi.de/>

```
[[tang1@jsfl01 pdaf]$ ll
total 102
-rw-r--r-- 1 tang1 jusers  482 Aug  6 18:53 First_steps.txt
-rw-r--r-- 1 tang1 jusers 7651 Aug  6 18:53 LICENSE.txt
-rw-r--r-- 1 tang1 jusers 11374 Aug  6 18:53 README
-rw-r--r-- 1 tang1 jusers  2422 Aug  6 18:53 README_PDAF-OMI
drwxr-xr-x 5 tang1 jusers  4096 Aug  6 18:53 external
drwxr-xr-x 2 tang1 jusers  4096 Aug  6 18:53 include
drwxr-xr-x 2 tang1 jusers  4096 Aug  6 18:53 lib
drwxr-xr-x 2 tang1 jusers  4096 Aug  6 18:53 make.arch
drwxr-xr-x 2 tang1 jusers 32768 Aug  6 18:53 src
drwxr-xr-x 7 tang1 jusers  4096 Aug  6 18:53 templates
```

Function: Do data assimilation.



hgs-pdaf source code

```
[[tang1@jsf102 offline_omi]$ ll
total 4797
-rw-r--r-- 1 tang1 jusers      5245 Apr 13  2023 Makefile
drwxr-xr-x 2 tang1 jusers     4096 Apr 13  2023 '_localization routines currently not needed'
-rw-r--r-- 1 tang1 jusers     4610 Apr 13  2023 assimilation_pdaf_offline.F90
-rw-r--r-- 1 tang1 jusers     5384 Aug 25 10:32 assimilation_pdaf_offline.o
-rw-r--r-- 1 tang1 jusers     8706 Apr 13  2023 callback_obs_pdafomi.F90
-rw-r--r-- 1 tang1 jusers     3744 Aug 25 10:32 callback_obs_pdafomi.o
-rw-r--r-- 1 tang1 jusers     1052 Apr 13  2023 collect_state_pdaf_offline.F90
-rw-r--r-- 1 tang1 jusers     1136 Aug 25 10:32 collect_state_pdaf_offline.o
drwxr-xr-x 2 tang1 jusers     4096 Apr 13  2023 dummympi
-rw-r--r-- 1 tang1 jusers      898 Apr 13  2023 finalize_pdaf.F90
-rw-r--r-- 1 tang1 jusers     1720 Aug 25 10:32 finalize_pdaf.o
-rw-r--r-- 1 tang1 jusers     1836 Apr 13  2023 g21_state_pdaf.F90
-rw-r--r-- 1 tang1 jusers     1456 Aug 25 10:32 g21_state_pdaf.o
-rwxr-xr-x 1 tang1 jusers 3754160 Aug 25 10:32 hgs-pdaf
-rw-r--r-- 1 tang1 jusers     2087 Apr 13  2023 init_dim_l_pdaf.F90
-rw-r--r-- 1 tang1 jusers     1120 Aug 25 10:32 init_dim_l_pdaf.o
-rw-r--r-- 1 tang1 jusers    12307 Apr 13  2023 init_ens_offline.F90
-rw-r--r-- 1 tang1 jusers    10648 Aug 25 10:32 init_ens_offline.o
-rw-r--r-- 1 tang1 jusers     1391 Apr 13  2023 init_n_domains_pdaf.F90
-rw-r--r-- 1 tang1 jusers     1280 Aug 25 10:32 init_n_domains_pdaf.o
```

Function: main program to carry out data assimilation for HGS.



User specified files

```
[[tang1@jsf102 HGS-PDAF_v1.0_GMD]$ ll
total 57
-rw-r--r-- 1 tang1 jusers 35149 Nov 20 21:52 LICENSE
-rw-r--r-- 1 tang1 jusers 3423 Nov 28 12:38 README.md
drwxr-xr-x 2 tang1 jusers 4096 Nov 28 13:30 docs
drwxr-xr-x 3 tang1 jusers 4096 Nov 28 11:41 examples
drwxr-xr-x 3 tang1 jusers 4096 Nov 20 21:52 hgs-pdaf
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 hgsiolib
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 namelists
drwxr-xr-x 4 tang1 jusers 4096 Nov 20 21:52 postprocessing
drwxr-xr-x 4 tang1 jusers 4096 Nov 20 21:52 preprocessing
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 run_scripts
```



Preprocessing tools

- The observation used for hgs-pdaf should be prepared in netCDF format
- This needs the use of executables provided in the *preprocessing* folder.

```
[[tang1@jsf102 preprocessing]$ ll
total 2
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 gen_obs_head
drwxr-xr-x 2 tang1 jusers 4096 Nov 20 21:52 gen_obs_sat
```

Function: Read the observation locations and values from HGS files, generate the .nc file which hgs-pdaf can read.



Preprocessing tools

- Example of generated netCDF file:

```
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 ;
```



Parameters settings

- Parameters, input and output path, file names are defined in the namelist files

```
[[tang1@jsf101 namelists]$ ll
total 1
-rw-r--r-- 1 tang1 jusers  402 Jun 23 10:38 namelist.hgs
-rw-r--r-- 1 tang1 jusers 1124 Jun 20 11:30 namelist.pdaf
```

Function: Define parameters and other configurations for HGS and data assimilation.



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'          Do not change!
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+
/

```



Namelist.pdaf - I

```
! Namelist for PDAF configuration
&pdaf
  ! General control
  str_daspec='da'      Prefix for the data assimilation output file
  filtertype=2          Assimilation method
  printconfig=.true.
  locweight=3           For localised filter, currently not available
  loctype=0             For inflation
  forget=1.0
  varscale=1.0
  type_trans=0
  type_forget=0
  use_global_obs = .true.
  istep=<pdafstep>
```



Namelist.pdaf - II

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

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

```
/
```

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

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

```
/
```

Namelist.pdaf - III

```
! Settings for assimilating HEAD observations
&pdaf_hgs
state_type = 6      Which variables will be included into the state vector
assim_o_head=.true.
rms_obs_head=0.05    Observation error
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'                                Full name of the observation file
file_head_suffix ='.nc'
```



Namelist.pdaf - IV

```
! 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>

/



Runscripts

- Python scripts to control the HGS runs
- Shell scripts to manage the assimilation runs

```
[tang1@jsf102 run_scripts]$ ll  
total 17  
-rwxr-xr-x 1 tang1 jusers 124 Nov 20 21:52 cpl_pdaf.inc.ucr  
-rwxr-xr-x 1 tang1 jusers 472 Nov 20 21:52 launch.sh  
-rwxr-xr-x 1 tang1 jusers 4785 Nov 20 21:52 runscript_n2.job
```

Function: control the assimilation run



HGS model run script – launch.sh

```
#!/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
#)
```



Control the assimilation run – shell script I

```
-rwxr-xr-x 1 tang1 jusers 5035 Aug  6 11:30 runscript_n2_jureca.job  
-rwxr-xr-x 1 tang1 jusers 5088 Aug  6 11:31 runscript_n2_jusuf.job  
-rwxr-xr-x 1 tang1 jusers 3682 Aug  6 11:31 runscript_n2_juwels.job  
-rwxr-xr-x 1 tang1 jusers 5510 Aug  6 11:40 runscript_n2_scicore.job
```

```
#!/bin/bash -x  
#SBATCH --account=icei-prace-2023-0004  
#SBATCH --nodes=1  
#SBATCH --ntasks=2  
#SBATCH --ntasks-per-node=2  
#SBATCH --output=2ens-%j.out  
#SBATCH --error=2ens-%j.err  
#SBATCH --time=00:10:00  
#SBATCH --partition=batch  
#SBATCH --mail-user=qi.tang@unine.ch  
#SBATCH --mail-type=ALL
```

For different supercomputing systems



Control the assimilation run – shell script II

```
# *** 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
```



Control the assimilation run – shell script III

```
# Ensemble 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 path 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
```



Control the assimilation run – shell script IV

```
# Ensemble run
# if firstrun=F

mkdir $ENS_DIR

for((ENS=1;ENS<=$NENS;ENS++))
do
    ENSstr=`printf %03d $ENS`      # ensemble size should be less than 999, otherwise
    echo $ENSstr
    export RUN_DIR=${ENS_DIR}${ENSstr}'/'          # output for all erstarts a
    mkdir $RUN_DIR
    cd $RUN_DIR
    ln -s $BASE_DIR/hgs-pdaf/offline_omi/hgs-pdaf
    cp $BASE_DIR/namelists/namelist.* .
    source ${BASE_DIR}/run_scripts/cpl_pdaf.inc.ucr
    cp namelist.pdaf temp.pdaf
    cp -r $MODEL_DIR/$ENSstr/* .
    chmod +x *.sh
done
```



Control the assimilation run – shell script V

```
cd $ENS_DIR

echo run started at realtime:
date
echo `pwd`

# create MPMD configuration file for different runs

if [ -e mpmd.conf ];then
    rm mpmd.conf
fi
touch mpmd.conf
```



Control the assimilation run – shell script VI

```
for((i=1;i<=$NENS;i++))  
do  
    ENSstr=`printf %03d $i`  
  
    echo '#!/bin/sh' > hgs-pdaf${ENSstr}  
    echo 'cd '${ENSstr} >> hgs-pdaf${ENSstr}  
    echo `pwd`/'${ENSstr}'/hgs-pdaf' >> hgs-pdaf${ENSstr}  
    chmod +x hgs-pdaf${ENSstr}  
  
    echo $((i-1))' ./hgs-pdaf'${ENSstr} >> mpmd.conf  
  
done  
cat mpmd.conf
```



Control the assimilation run – shell script VII

```
if [ -e mpmd-hgs.conf ];then
    rm mpmd-hgs.conf
fi
touch mpmd-hgs.conf

for((i=1;i<=$NENS;i++))
do
    ENSstr=`printf %03d $i`

    echo '#!/bin/sh' > model${ENSstr}
    echo 'cd ${ENSstr} >> model${ENSstr}'
    echo `pwd`'/'${ENSstr}'/launch.sh' >> model${ENSstr}
    chmod +x model${ENSstr}

    echo $((i-1))' ./model'${ENSstr} >> mpmd-hgs.conf

done
cat mpmd-hgs.conf
```



Control the assimilation run – shell script VIII

```
if [ -e mpmd-spinup.conf ];then
  rm mpmd-spinup.conf
fi
touch mpmd-spinup.conf

for((i=1;i<=$NENS;i++))
do
  ENSstr=`printf %03d $i`

  echo '#!/bin/sh' > spinup${ENSstr}
  echo 'cd '${ENSstr}' >> spinup${ENSstr}
  echo `pwd`'/'${ENSstr}'/spinup.sh' >> spinup${ENSstr}
  chmod +x spinup${ENSstr}

  echo $((i-1))' ./spinup'${ENSstr}' >> mpmd-spinup.conf

done
cat mpmd-spinup.conf
```



Control the assimilation run – shell script VIII

```
# Check if this is the first time step. If yes, run spinup; if no, skip
if [[ $firstrun = "T" ]]; then
    for((ENS=1;ENS<=$NENS;ENS++))
    do
        ENSstr=`printf %03d $ENS`      # ensemble size should be less than 99
        echo $ENSstr
        export RUN_DIR=${ENS_DIR}${ENSstr}'/'          # output for all runs
        cd $RUN_DIR
        ln -s $HGS_DIR/grok HGS/grok
        ln -s $HGS_DIR/phgs HGS/hgs
        ln -s $HGS_DIR/hsplot HGS/hsplot
    done
    cd $ENS_DIR
    srun -l --propagate=STACK --multi-prog mpmd-spinup.conf &> spinup.out
    firstrun=F
fi
```



Control the assimilation run – shell script X

```
# time counter
for((pdafstep=1;pdafstep<=$TSTEPS;pdafstep++))
do
  export step_counter=$pdafstep
  echo $step_counter
  for((ENS=1;ENS<=$NENS;ENS++))
  do
    ENSstr=`printf %03d $ENS`      # ensemble size should be less than 999, otherwise h
    echo $ENSstr
    export RUN_DIR=${ENS_DIR}${ENSstr}'/'          # output for all erstarts and
    cd $RUN_DIR
    cp temp.pdaf namelist.pdaf
    sed -e "s@<pdafstep>@${step_counter}@g" \
      namelist.pdaf > toto
    mv toto namelist.pdaf
    cp HGS/Grokfiles/wells/wells_flow_${step_counter}.inc HGS/Grokfiles/wells_flow.inc
  done
  cd ${ENS_DIR}
  srun -l --propagate=STACK --multi-prog mpmd-hgs.conf &> hgs.out.${step_counter}
  srun -l --propagate=STACK --multi-prog mpmd.conf &> hgs-pdaf.out.${step_counter}
  for((ENS=1;ENS<=$NENS;ENS++))
  do
    ENSstr=`printf %03d $ENS`
    echo $ENSstr
    export RUN_DIR=${ENS_DIR}${ENSstr}'/'
    cd $RUN_DIR
    cp HGS/Flowo.head_pm.0001 HGS/IC/Ini_pm
    cp HGS/Flowo.head_olf.0001 HGS/IC/Ini_olf
  done
  cd ${ENS_DIR}
done
```



Post-processing tools

- Python scripts to analyse and visualise the results

```
[[tang1@jsf103 postprocessing]$ ll
total 2
drwxr-xr-x 2 tang1 jusers 4096 Aug  6 11:43 head
drwxr-xr-x 2 tang1 jusers 4096 Aug  6 11:43 sat
```

```
[[tang1@jsf103 head]$ ll
total 21
-rw-r--r-- 1 tang1 jusers 3397 Aug  6 11:43 cal_diff_fore-obs_pump.py
-rw-r--r-- 1 tang1 jusers 3014 Aug  6 11:43 cal_rmse_fore.py
-rw-r--r-- 1 tang1 jusers 1080 Aug  6 11:43 plot.py
-rw-r--r-- 1 tang1 jusers 2926 Aug  6 11:43 plot_absdiff_ana-obs.py
-rw-r--r-- 1 tang1 jusers 2308 Aug  6 11:43 plot_absdiff_fore-obs.py
-rw-r--r-- 1 tang1 jusers 2254 Aug  6 11:43 plot_diff_ana-obs.py
-rw-r--r-- 1 tang1 jusers 2254 Aug  6 11:43 plot_diff_fore-obs.py
-rw-r--r-- 1 tang1 jusers 3503 Aug  6 11:43 plot_mean_absdiff_ana-obs.py
-rw-r--r-- 1 tang1 jusers 3503 Aug  6 11:43 plot_mean_absdiff_fore-obs.py
-rw-r--r-- 1 tang1 jusers 2183 Aug  6 11:43 plot_obs_ana.py
-rw-r--r-- 1 tang1 jusers 4362 Aug  6 11:43 plot_obs_fore.py
```



A screenshot of a web browser showing the GitHub Discussions page for the repository "HGS-PDAF_v1.0_GMD". The browser interface includes standard window controls (red, yellow, green circles), a back/forward button, a refresh icon, and a search bar with placeholder text "Type ⌘ to search". The address bar shows the URL "github.com". The main navigation bar at the top has links for "Code", "Issues", "Pull requests", "Discussions" (which is underlined in red), "Actions", "Projects", "Wiki", "Security", "Insights", and "Settings". Below the navigation bar is a search bar with the query "is:open", followed by filters for "Sort by: Latest activity", "Label", "Filter: Open", and a green "New discussion" button. The left sidebar is titled "Categories" and lists "View all discussions" (selected), "Announcements", "General", "Ideas", "Polls", "Q&A", and "Show and tell", each with a corresponding emoji icon. The main content area features a "Welcome to discussions!" message with a speech bubble icon, stating: "Discussions are to share announcements, create conversation in your community, answer questions, and more. To get started, you can create a [new discussion](#)". A callout box titled "Most helpful" encourages users to mark answers: "Be sure to mark someone's comment as an answer if it helps you resolve your question — they deserve the credit! ❤️". At the bottom of the sidebar are links for "Community guidelines" and "Community insights".

Discussion on GitHub

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

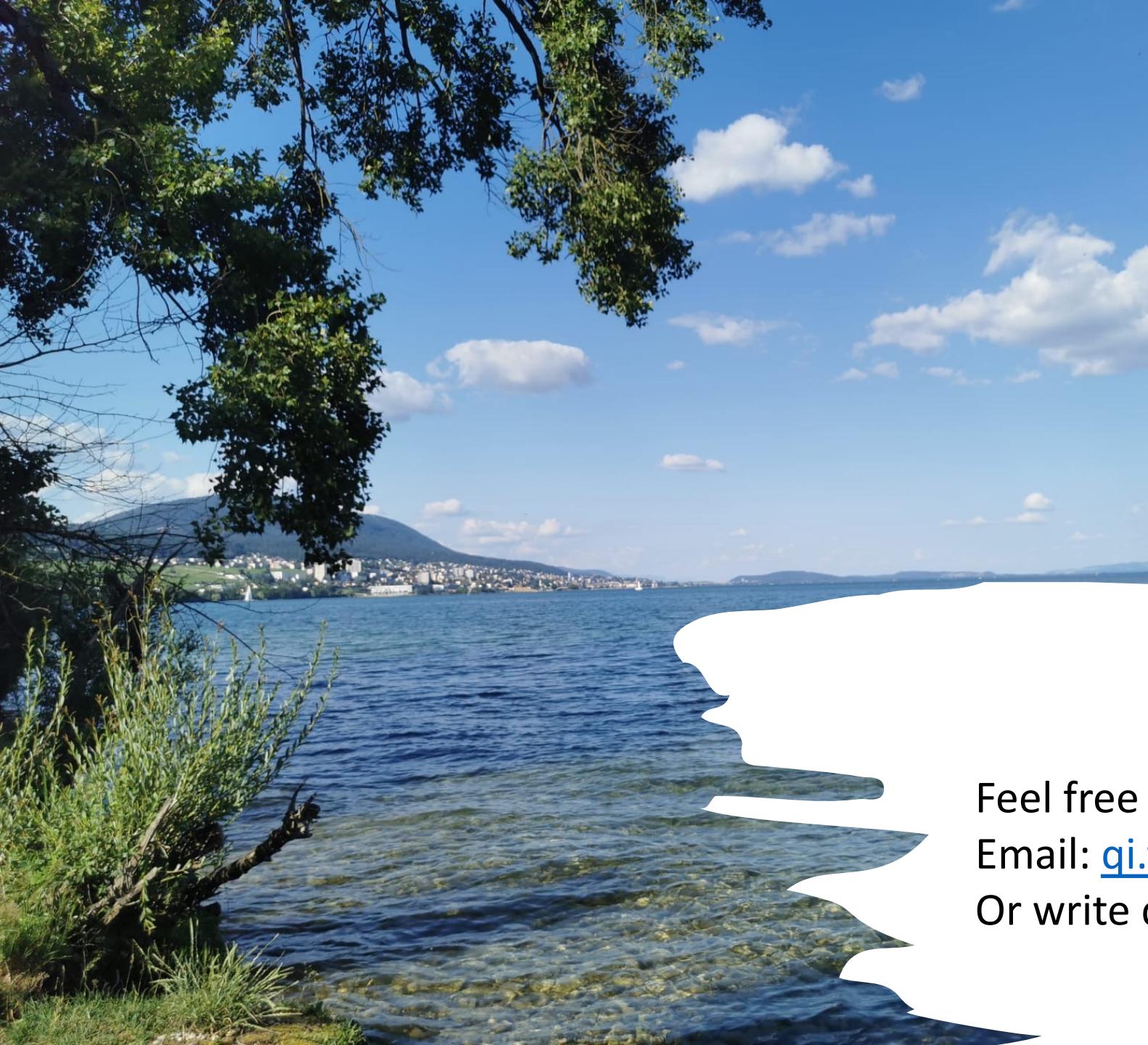
For advanced user:

- 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





Enjoy it!

Feel free to contact us if you need any help:
Email: qi.tang@unine.ch
Or write directly on GitHub!