

# Deltares

Enabling Delta Life



DFMWAQ with Python

# Agenda

---

- D-Flow with processes (DFMWAQ) use case
- Why might you want to use scripting?
- Arranging the input data
- Making the model
- Making a restart file
- Examining the balances
- Making coastal water quality boundaries (if there is time)
- Remember: Still beta!
- These python tools are not a Deltares product



# Use case

Total time for simulation →

- In the past

Hydro model

WQ model

- Common now

Hydro model

WQ model

Parallelizability ↓

# D-FLOW with processes – parallelizing water quality simulations

- **Pro:** no more big water balance files
  - **Pro:** *no other coupling problems (e.g. changes in cell numbers in D-WAQ compared to D-FLOW)*
  - **Pro:** no parallelized transport solver in D-WAQ needed
  - **Con:** rerunning hydrodynamics is expensive (in CPU time)
- 
- It is common that you might want to **compare the speed** of an **online** model with that of an **offline** model
  - For this you need to be able to **create the same model twice** using quite different architectures
  - It is best to do this in a structured way

# Typical workflow to consider

---

1. The user has a D-FLOW hydrodynamic model that they want to use as the foundation of a water quality model
2. Run the FLOW model, obtain coupling output files, and import into DeltaShell
3. Set up the water quality simulation and run D-WAQ

## What if you want to see if D-FLOW with processes is a good choice for your application??

Option 1:

- Augment the D-FLOW model in Deltashell to include processes (shown in previous workshop)

Option 2:

- ‘Grow’ a new model using the old D-FLOW model and a sub file
- This is what we will look at today

# 'Grow' a D-FLOW with processes model

- Model Prerequisites:
  - A D-FLOW model
  - A sub file
  - Deltashell-format boundary condition files (optional)
- These will be provided for this workshop
- Python Prerequisites
  - Download the following repository: <https://github.com/RSchueder/dflowutil>
  - Follow readme instructions

## dflowutil

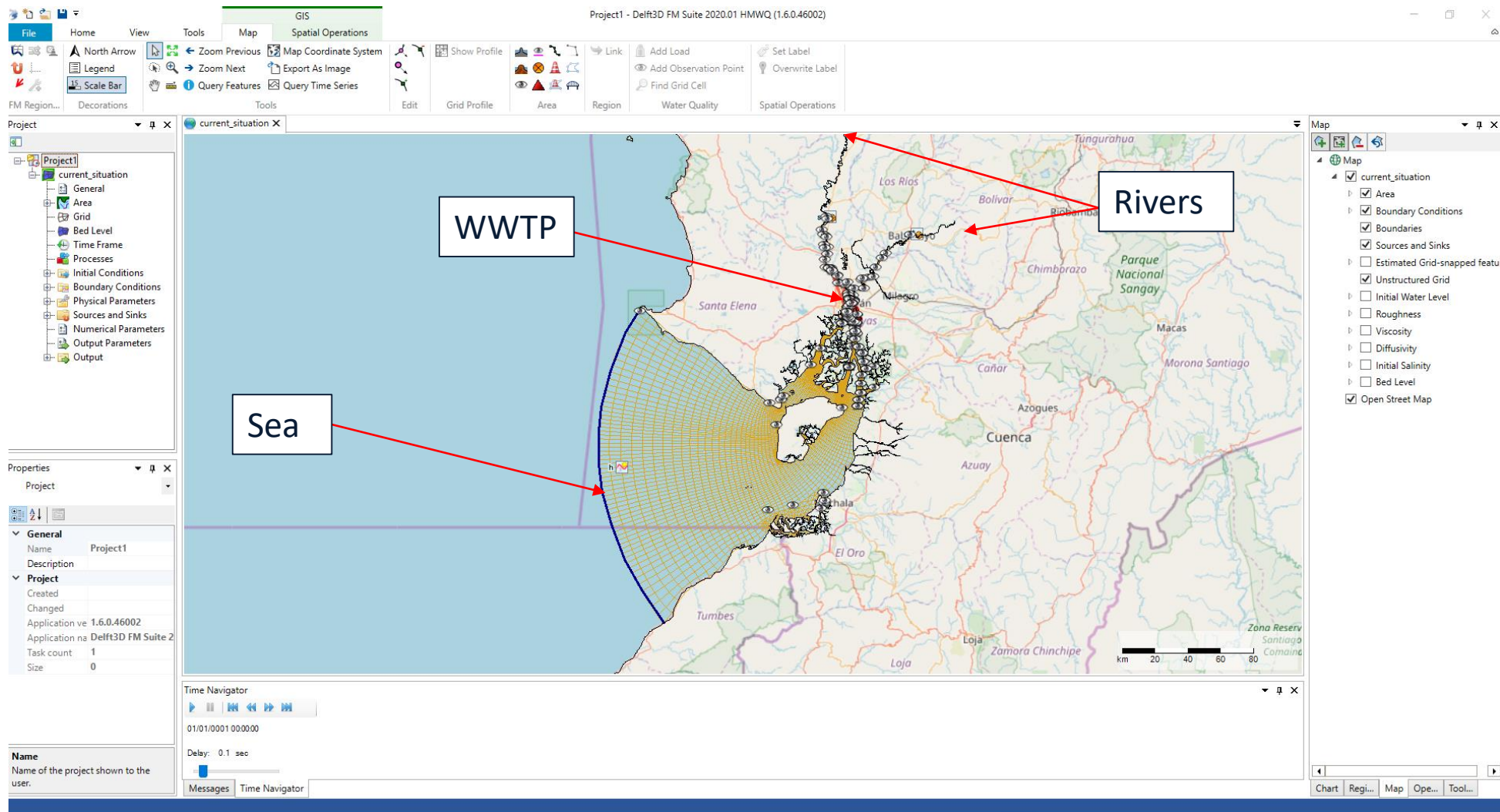
Utilities for Delft3D FLOW Flexible Mesh in python. It contains utilities useful for both hydrodynamic and integrated water quality simulations.

### INSTALL

- \$ python setup.py sdist bdist\_wheel
- \$ (optional) conda env create -f dflowutil.yml
- \$ (optional) conda activate dflowutil
- \$ pip install dist/dflowutil-0.1.3-py3-none-any.whl

tests.py provides an overview of key functionality

# The FLOW model we will be working with



Open this  
model in  
deltashell

Deltares

# dflowutil.DFMWAQModel

- dflowutil is a python *package*
- DFMWAQModel is a *module* of the package
- It is a class that holds information and functions needed to build a new D-FLOW with processes model from a D-FLOW model

```
class DFMWAQModel():

    def __init__(self, mdu, ext, subfile, new_dir, tref, ini=None, run_sys='linux', v=None, cores=None,
                 loads_data=None, bounds_data=None,
                 process_path=None):
        """
        DFMWAQ model initialized from DFM model inputs

        call DFMWAQModel.build() to build an initialized model
        Arguments:
            mdu {str} -- path to mdu
            ext {list} -- [ext1, ext2], if only one, still must be a list
            subfile {dflowutil.SubFile} -- a dflowutil.SubFile object
            new_dir {path} -- path where model will be built
            tref {datetime.datetime}
            ini {dict} -- sub name value pair for initial conditions. May be empty.
            v {str} -- version - i.e. 1.2.56.xxx.
            cores {list} -- [nodes, threads]
        """
```



# What does dflowutil.DFMWAQModel do?

- It builds an entire model from a sub file. This way there is no room for mistake in the ordering of substances in the loads file!
- It scans the mdu for all model assets, excluding boundary conditions (specify these explicitly)
- It creates boundary conditions for all transportable substances. The data is either dummy ( $= 0$ ) or is taken from deltashell-formatted \*.csv files (if available)
- It creates initial conditions for all substances, either uniform, or as 2D field

```
def build(self):
    """
    builds model based solely on attributes from initialization
    """
    self.soursin()
    self.open_bnd()
    self.administrate_ext_files()
    self.merge_ext_files()
    self.run_set_file()
```

# How can I use it: making the model

```
#####
...
model with boundary data
...

process_path= r'd:\projects\dflowutil\tests\DSD\01_substances\proc_def.dat'
system = 'windows'
mdu = r'd:\projects\dflowutil\tests\DSD\00_src\current_situation.mdu'
sub_file = r'd:\projects\dflowutil\tests\DSD\01_substances\guayas_V11.sub'

# boundaries for the mdu
ext = [r'd:\projects\dflowutil\tests\DSD\00_src\plant_loads_current_local.ext',
      r'd:\projects\dflowutil\tests\DSD\00_src\sea_riv_boundary_local_bc.ext']

# water quality data
loads_data = [r'd:\projects\dflowutil\tests\DSD\03_loads\full_model_ds.csv']
bounds_data = [r'd:\projects\dflowutil\tests\DSD\02_boundaries\river_ds.csv',
              r'd:\projects\dflowutil\tests\DSD\02_boundaries\sea_ds.csv']

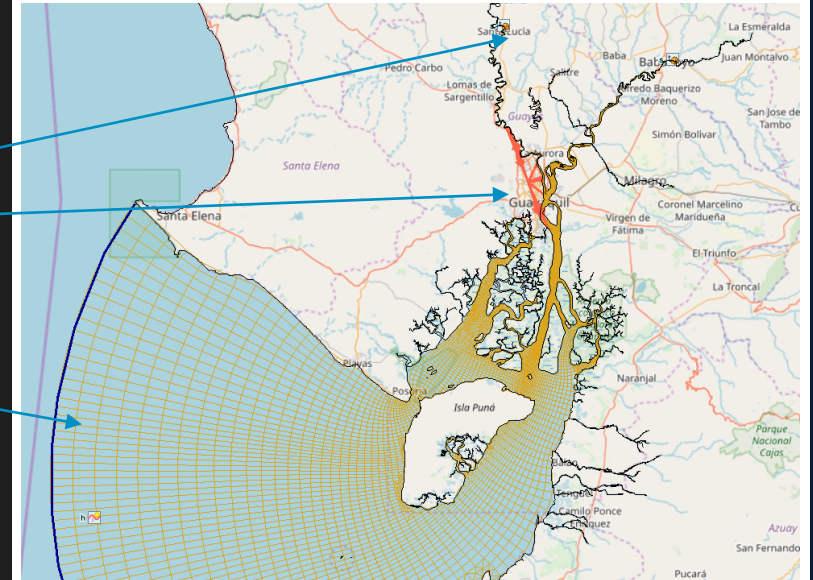
# location of new model
new_model_dir = 'd:\\projects\\dflowutil\\tests\\DSD\\R01\\'

# sub file to use
substances = SubFile(sub_file)

# initial conditions
ini = {'OXY' : 7}
tref = datetime.datetime(2000, 1, 1)
ecuador_model = DFMWAQModel(mdu, ext, substances, new_model_dir, tref,
                             bounds_data = bounds_data, loads_data = loads_data,
                             run_sys = system, process_path = process_path)
ecuador_model.build()

#####
```

Task for you:  
Please edit the  
paths



# Set up conda environment and running

---

```
Anaconda Prompt (Anaconda3)

(base) C:\Users\schueder>d:
(base) D:\>cd projects\dflowutil
(base) D:\projects\dflowutil>conda env create -f dflowutil.yml
CondaValueError: prefix already exists: D:\Anaconda3\envs\dflowutil

(base) D:\projects\dflowutil>conda activate dflowutil
(dflowutil) D:\projects\dflowutil> python make_model.py
```

# Examine the new model

- d:\projects\dfLOWUTIL\tests\DSD\R01\

[.]

[DFM\_OUTPUT\_current\_situation]

run\_model

current\_situation\_dry

stage4\_spot\_ortho\_DFM\_reint\_net

guayas

bridgepiers\_3\_thd

guayas\_locations\_obs

proc\_def

proc\_def

bloom

bloomingp

current\_situation

L\_Orquideas

L\_Orquideas

bat
xyz
nc
ldb
pli
xyn
dat
def
spe
d09
mdu
pli
tim

<DIR> 11-11-2019 16:40 ----

<DIR> 11-11-2019 16:48 ----

189 07-11-2019 12:17 -a--

4,400 11-11-2019 16:40 -a--

6,367,500 11-11-2019 16:40 -a--

3,844,515 11-11-2019 16:40 -a--

1,170 11-11-2019 16:40 -a--

10,340 11-11-2019 16:40 -a--

2,508,780 11-11-2019 16:40 -a--

42,652 11-11-2019 16:40 -a--

53,593 11-11-2019 16:40 -a--

1,208 11-11-2019 16:40 -a--

16,595 11-11-2019 16:40 -a--

125 11-11-2019 16:40 -a--

65,234 11-11-2019 16:40 -a--

[processes]
SubstanceFile = d:\projects\dfLOWUTIL\tests\DSD\01\_substances\guayas\_V11.sub
AdditionalHistoryOutputFile =
ProcesDataBaseFile = proc\_def.dat
DtProcesses = 600. # waq processes time step
DtMassBalance = 86400. # waq mass balance output time step
ProcessFluxIntegration = 1 # integration option (1: WAQ, 2: D-Flow FM)

# Running the model

- We need to edit a batch file to allow for running in this exercise
- C:\Program Files (x86)\Deltares\Delft3D FM Suite 2020.01 HMWQ (1.6.0.46002)\plugins\DeltaShell.Dimr\kernels\x64\dflowfm\scripts\run\_dflowfm\_processes.bat



```
@ echo off
title %cd%
rem
rem This script is an example for running DFlowFM on Windows
rem Adapt and use it for your own purpose
rem
rem
rem Set the directories
rem
set scriptdir=%~dp0
set dflowfmexedir=%scriptdir%\..\bin
set libdir=%scriptdir%\..\..\share\bin
set procdefbloomspedir=%scriptdir%\..\default
set path=%dflowfmexedir%;%libdir%;%path%

rem
rem Run
rem
"%dflowfmexedir%\dflowfm-cli.exe" --nodisplay --autostartstop %1 %2 %3 %4 %5 %6 %7 %8 --processlibrary "%procdefbloomspedir%\proc_def.dat" --bloomspecies "%procdefbloomspedir%\bloom.spe"
```

Ensure there are quotes here

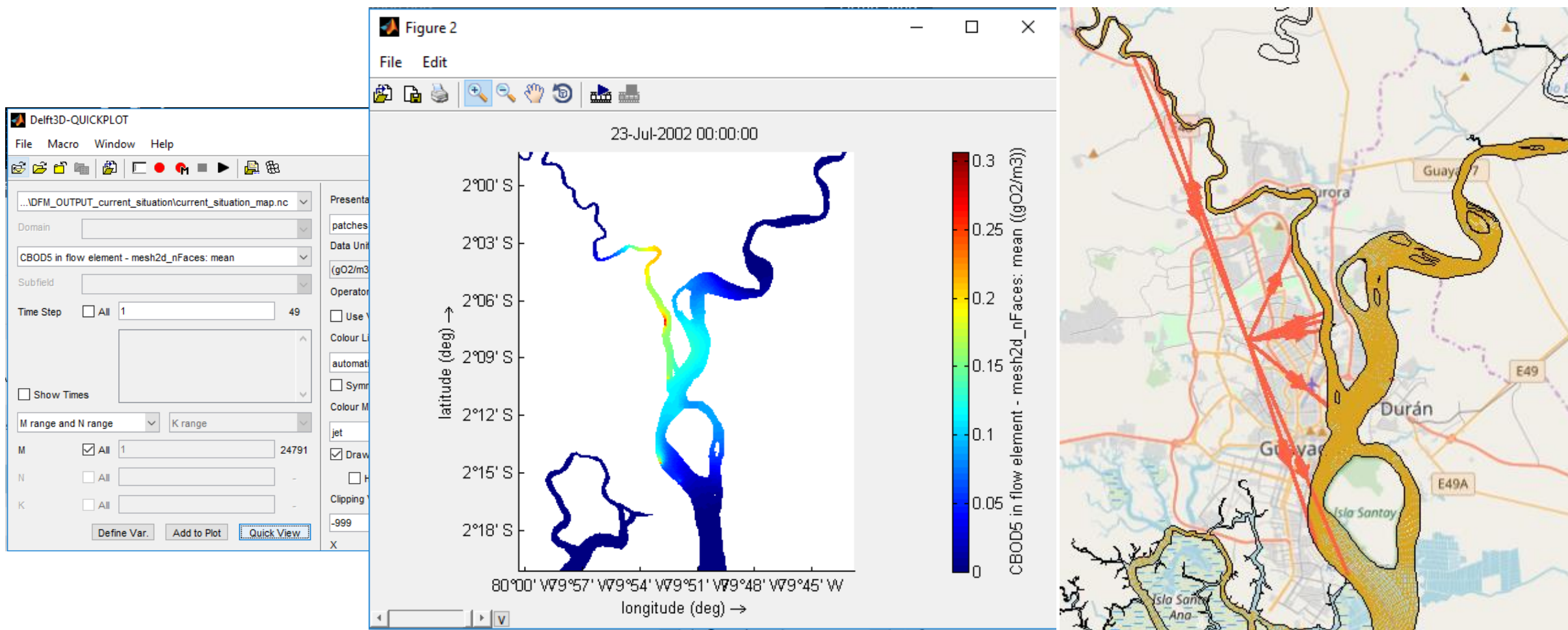
# Run and take a break

▼ d:\projects\dfLOWUTIL\tests\DSD\*,*				
Name	Ext	Size	↑ Date	Attr
↑ [..]		<DIR>	11-11-2019 17:12	----
[00_src]		<DIR>	11-11-2019 16:36	----
[01_substances]		<DIR>	07-11-2019 16:15	----
[02_boundaries]		<DIR>	01-11-2019 09:55	----
[03_loads]		<DIR>	04-11-2019 13:45	----
[04_obs]		<DIR>	11-11-2019 17:13	----
[R01]		<DIR>	11-11-2019 16:40	----
[R02]		<DIR>	11-11-2019 16:54	----
run_model	bat	189	07-11-2019 12:17	-a--

Copy the bat file to the run directory

▼ d:\projects\dfLOWUTIL\tests\DSD\R01\*,*				
Name	Ext	Size	↑ Date	Attr
↑ [..]		<DIR>	11-11-2019 16:40	----
[DFM_OUTPUT_current_situation]		<DIR>	11-11-2019 16:48	----
run_model	bat	189	07-11-2019 12:17	-a--
current_situation_dry	xyz	4,400	11-11-2019 16:40	-a--
stage4_spot_ortho_DFM_reint_net	nc	6,367,500	11-11-2019 16:40	-a--

# Take a look at the results



# Making a restart

```
new_model_dir = 'd:\\projects\\dflowutil\\tests\\DSD\\R02\\'
```

\dflowutil\make\_model.py (you ran this earlier)

```
...
turn the last model into a restart
...

sub_file = r'd:\projects\dflowutil\tests\DSD\01_substances\guayas_V11.sub'
map_dir = 'd:\\projects\\dflowutil\\tests\\DSD\\R01\\DFM_OUTPUT_current_situation\\'
new_model_dir = 'd:\\projects\\dflowutil\\tests\\DSD\\R02\\'
rst_to_xyz(map_dir, sub_file, -1, new_model_dir)
```

\dflowutil\make\_restart.py

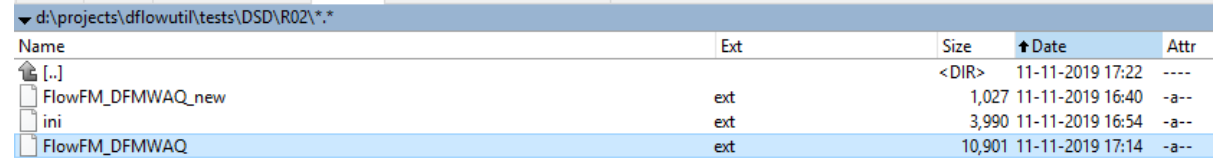
Anaconda Prompt (Anaconda3)

```
(base) C:\Users\schueder>d:
(base) D:\>cd projects\dflowutil
(base) D:\projects\dflowutil>conda env create -f dflowutil.yml
CondaValueError: prefix already exists: D:\Anaconda3\envs\dflowutil

(base) D:\projects\dflowutil>conda activate dflowutil
(dflowutil) D:\projects\dflowutil> python make_restart.py
```



# Adding the restart to the model



Name	Ext	Size	↑ Date	Attr
<DIR>			11-11-2019 17:22	----
FlowFM_DFMWAQ_new	ext	1,027	11-11-2019 16:40	-a--
ini	ext	3,990	11-11-2019 16:54	-a--
FlowFM_DFMWAQ	ext	10,901	11-11-2019 17:14	-a--

- Open R02/FlowFM\_DFMWAQ.ext
- Delete old initial conditions from bottom of file
- Open R02/ini.ext
- Copy contents from ini.ext -> FlowFM\_DFMWAQ.ext
- In FlowFM\_DFMWAQ.ext , rename:
  - initialtracers1 -> initialwaterlevel
  - initialtracersa1 -> initialsalinity
- Delete from top of FlowFM\_DFMWAQ.ext :

```
QUANTITY=initialwaterlevel
FILENAME=ini_waterlevel_20020722.xyz
FILETYPE=7
METHOD=5
OPERAND=O

QUANTITY=initialsalinity
FILENAME=ini_salinity_20020722.xyz
FILETYPE=7
METHOD=5
OPERAND=O
```

# Adding a balance area to the model

- Good idea to include balances so you can examine the relative contribution of different loads

```
QUANTITY=waqmassbalanceareaDomain
FILENAME=domain.pol
FILETYPE=10
METHOD=4
OPERAND=0
VALUE=1.0
```

Add to -> \dflowutil\tests\DSD\R02\FlowFM\_DFMWAQ.ext

- The domain.pol file was created for you when the model was created.
- It is a polygon that contains the entire domain

# Run and take a break

▼ d:\projects\dfLOWUTIL\tests\DSD\\*,\*

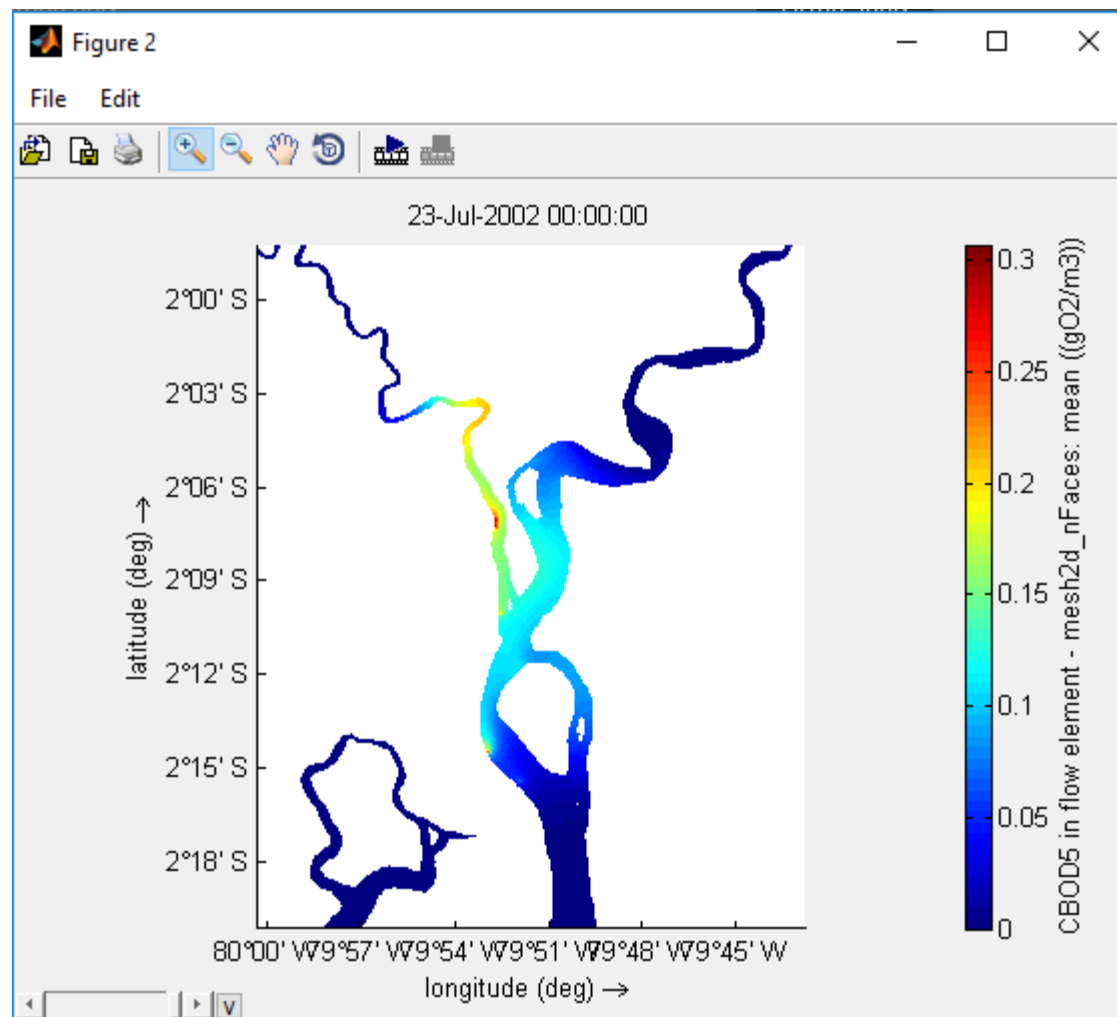
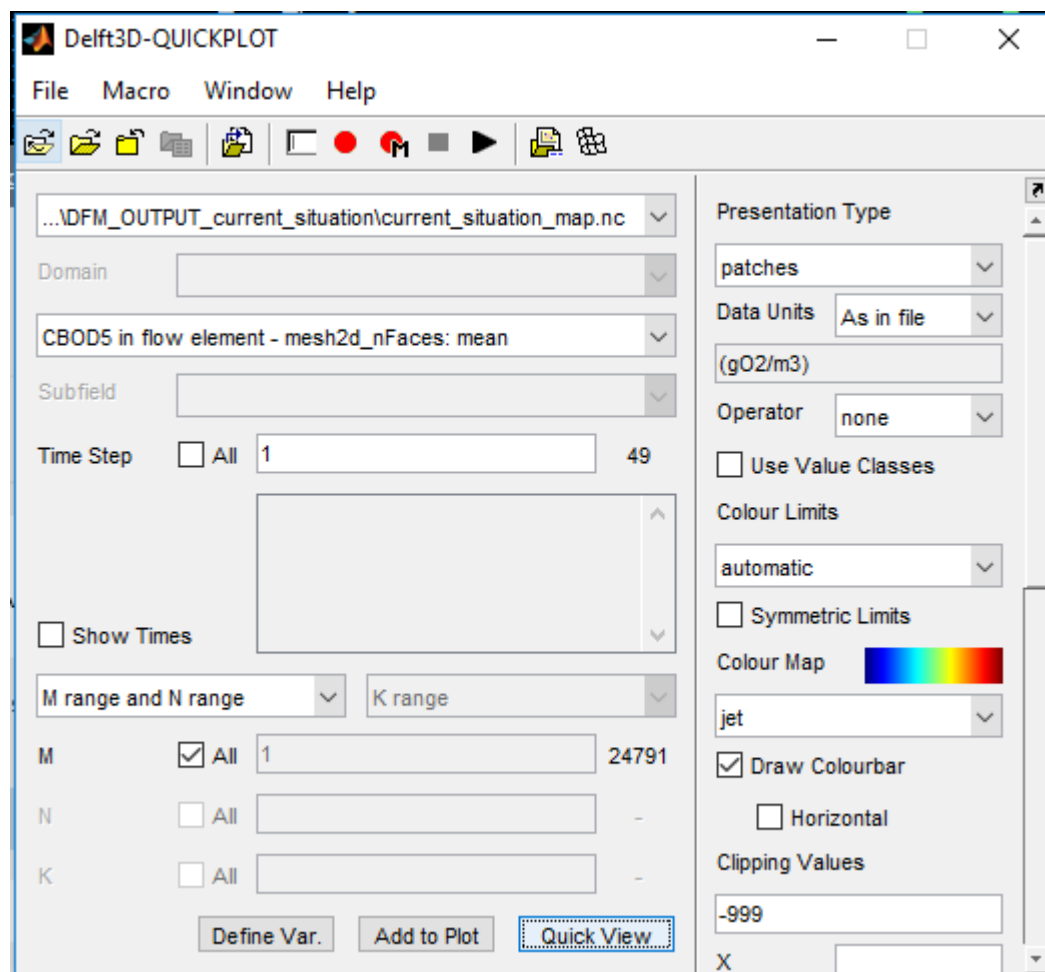
Name	Ext	Size	↑Date	Attr
⬆ [..]		<DIR>	11-11-2019 17:12	----
[00_src]		<DIR>	11-11-2019 16:36	----
[01_substances]		<DIR>	07-11-2019 16:15	----
[02_boundaries]		<DIR>	01-11-2019 09:55	----
[03_loads]		<DIR>	04-11-2019 13:45	----
[04_obs]		<DIR>	11-11-2019 17:13	----
[R01]		<DIR>	11-11-2019 16:40	----
[R02]		<DIR>	11-11-2019 16:54	----
run_model	bat	189	07-11-2019 12:17	-a--

Copy the bat file to the run directory

▼ d:\projects\dfLOWUTIL\tests\DSD\R02\\*,\*

Name	Ext	Size	↑Date	Attr
⬆ [..]		<DIR>	11-11-2019 17:22	----
[DFM_OUTPUT_current_situation]		<DIR>	11-11-2019 17:22	----
run_model	bat	189	07-11-2019 12:17	-a--
stage4_spot_ortho_DFM_reint_net	nc	6,367,500	11-11-2019 16:40	-a--

# Take a look at the results



# Create balances

---

```
'''
examine balances
'''
sub_file = r'd:\projects\dflowutil\tests\DSD\01_substances\guayas_V11.sub'
bal_file = r'd:\projects\dflowutil\tests\DSD\R02\DFM_OUTPUT_current_situation\current_situation_wq_proc_bal.txt'
subs = SubFile(sub_file)
prn = BalanceFile(bal_file)
area = ['Domain']
period = ['933-934']
prn.extract_balances(area, period)
```

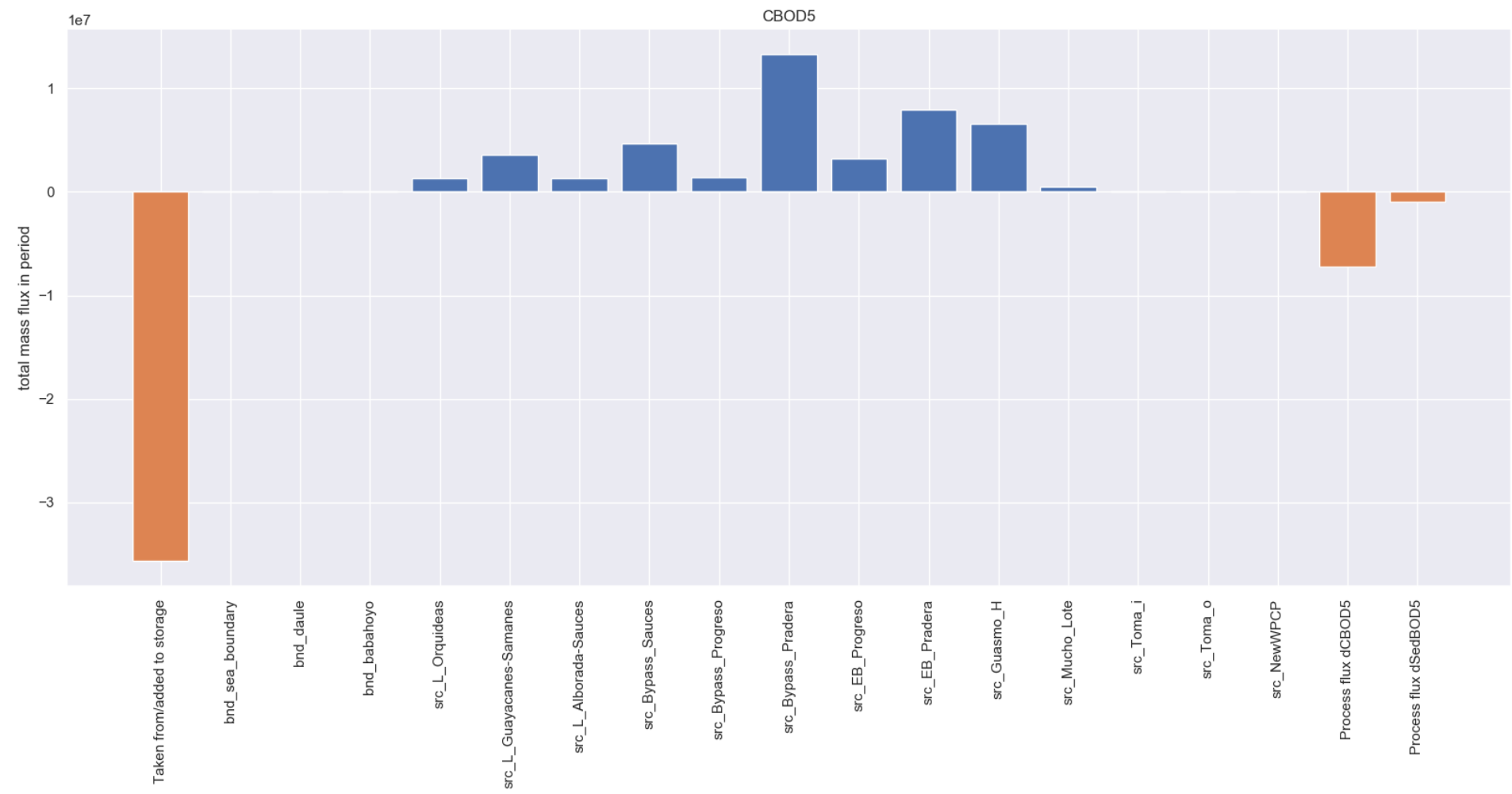
# Create balances

```
...
now do some plotting
...

df = prn.areas
subs = SubFile(sub_file)
for domain in area:
    plt.close('all')
    for sub in subs.substances:
        if subs.transportable[sub] == 'active':
            fig = plt.figure()
            ax = fig.add_axes([0.1,0.3,0.8,0.6])
            print(sub)
            dfi = prn.areas[domain][sub]['Inflows']
            dfo = prn.areas[domain][sub]['Outflows']

            x = np.arange(0, len(dfi.columns))
            ax.bar(x, dfi.iloc[0])
            ax.bar(x, -dfo.iloc[0])
            ax.set_ylabel('total mass flux in period')
            plt.title(sub)
            plt.xticks(x, dfi.columns,rotation=90)
```

# Examine balances



# Conclusion

---

- We started with a D-Flow FM hydrodynamic model and decided to add water quality processes to it
- We obtained a substance file, and some Deltashell-formatted csv files
- We built a new D-Flow FM model from the old one, with new boundaries and a new [processes] section in the mdu
- We ran the model and examined the output
- We created restart samples from one model and ran a new model using this initial condition
- We examined the computed mass balances for different substances

## Notes:

- Does not support 3D boundaries yet. See [https://github.com/RSchueder/COASTSERV\\_Model/tree/FlaskApp](https://github.com/RSchueder/COASTSERV_Model/tree/FlaskApp) for examples of how to do this using CMEMS data
- Did not cover things like spatial parameters (critical shear stress) or temporal functions (i.e. solar radiation). These are possible to use!
- Not extensively tested, not an official Deltares product, collaboration is welcome!



# Any questions?





## Making the DFMWAQ model

[illegible]