Deltares

dfm_tools

A Python package for D-FlowFM modelbuilding and post-processing

Jelmer Veenstra

14-11-2023

dfm_tools: the concept

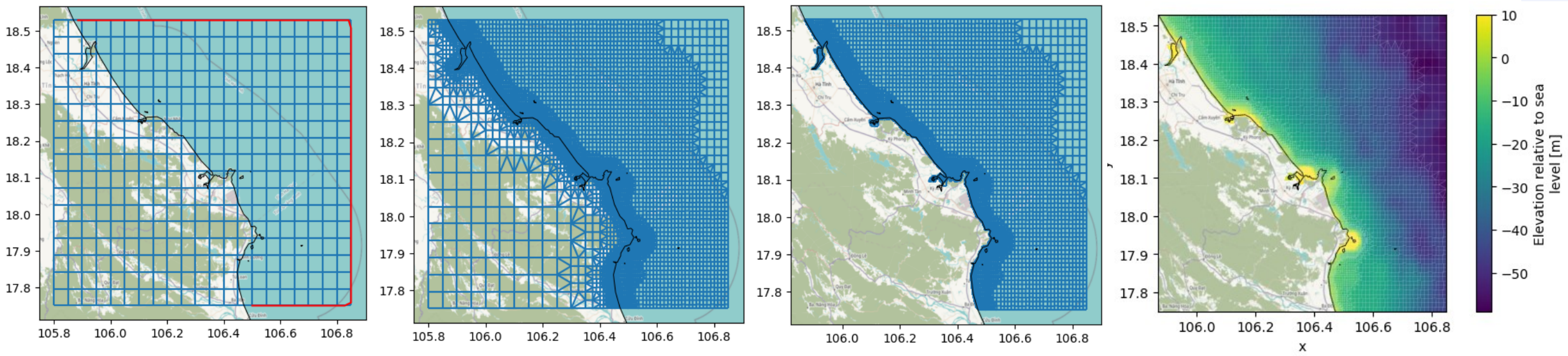
- What:
  - a Python package for D-FlowFM modelbuilding and post-processing
  - contains convenience functions built on top of other packages like xarray, xugrid, HYDROLIB-core, MeshkernelPy and many more
  - sharing of workflows among users (colleagues/external)
  - no need to re-invent the wheel

- Where:
  - code on github: https://github.com/Deltares/dfm_tools
  - installable via pip
  - jupyter notebooks with example code

Pre-processing: Modelbuilder

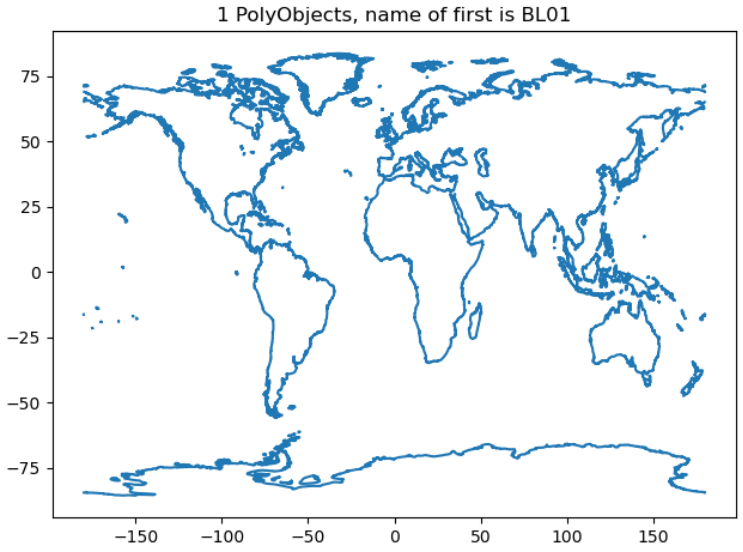
Grid generation with MeshKernelPy

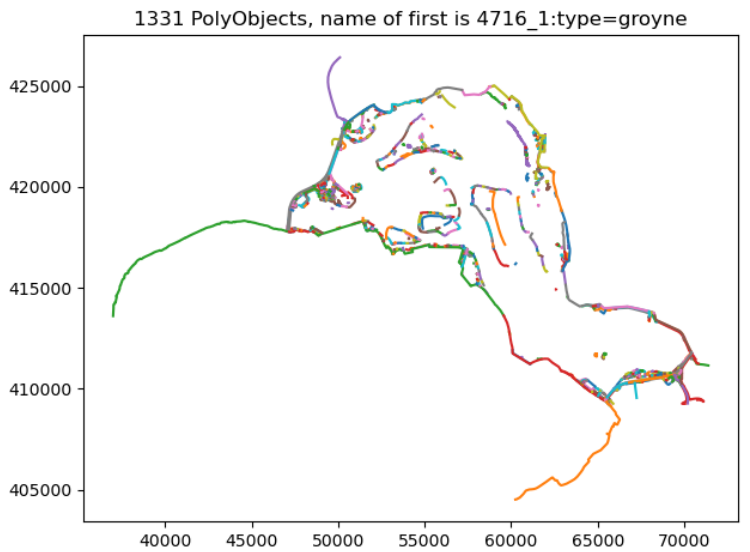
- **MeshKernelPy**: A Python package wrapping MeshKernel (meshing algorithms in C++)
- Generation of grids: regular, refinement, deletion, interpolation
- Multiple mesh instances: Mesh1D, Mesh2D, Contacts

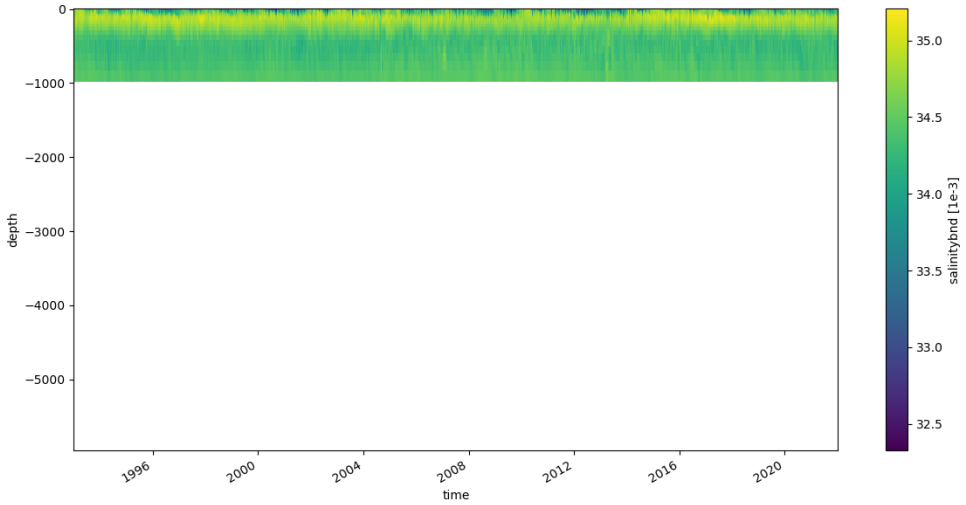


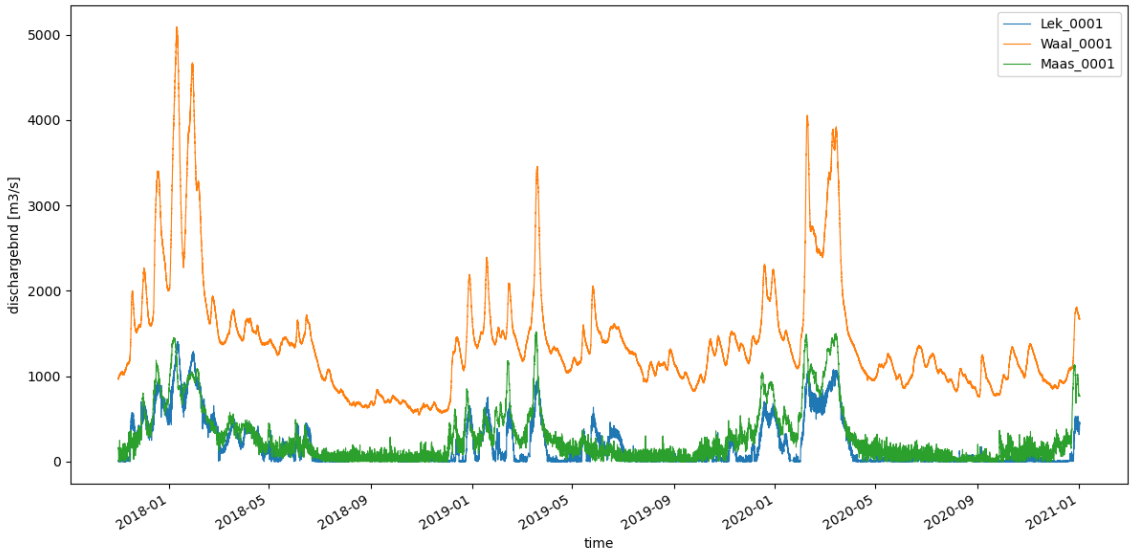
Reading + plotting of D-FlowFM input files

- HYDROLIB-core for i/o of D-FlowFM model input files
- conversion to xarray/pandas datasets with dfm_tools
- figures: pol/pli/pliz/ldb and bc files





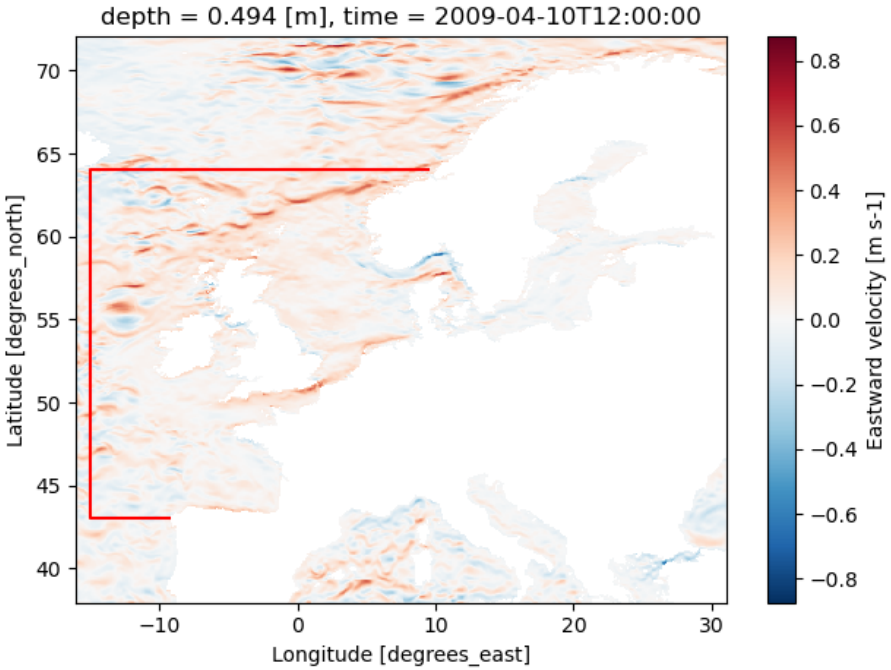


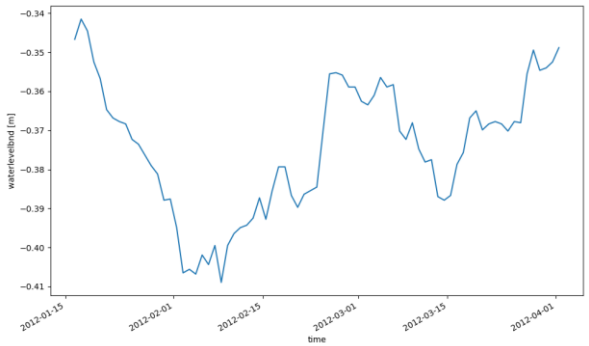


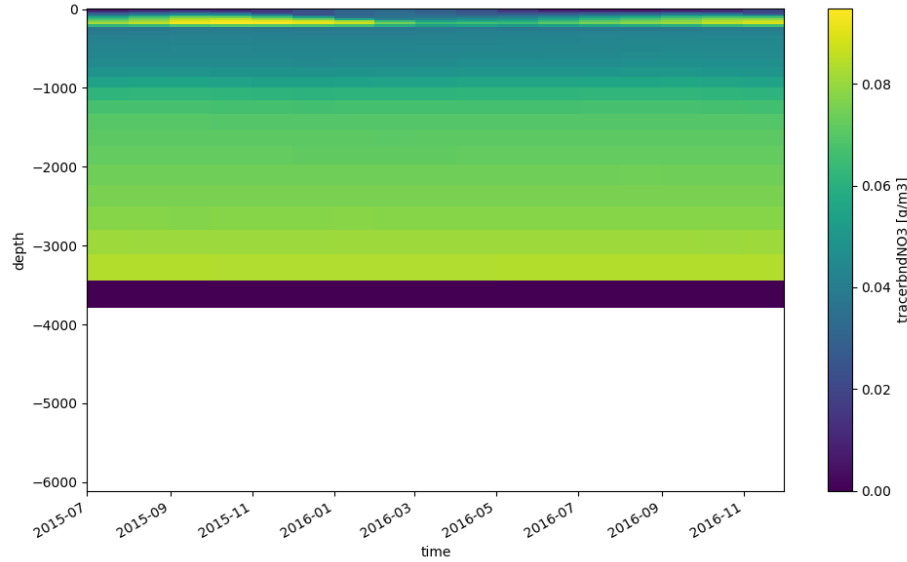
Deltares

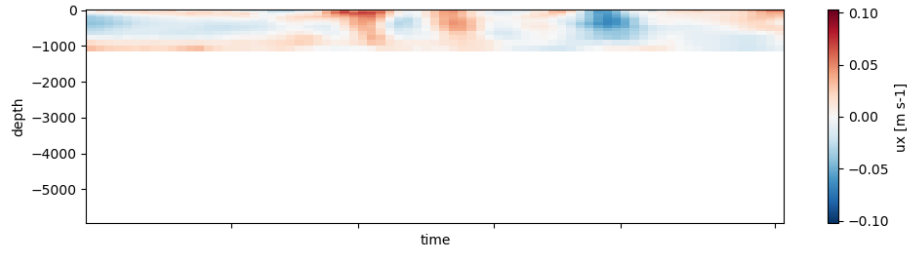
Interpolation of ocean models to polyline (.bc files)

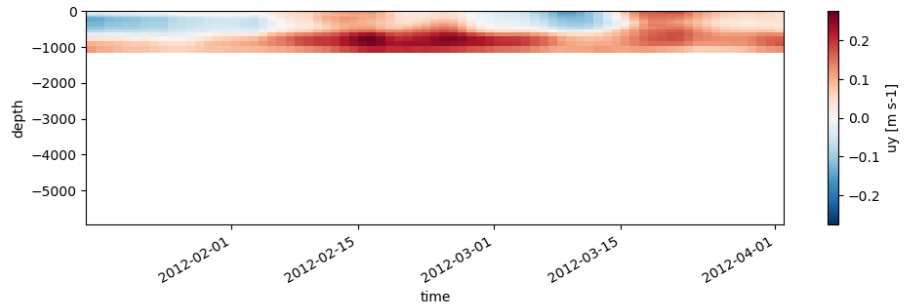
- Download ocean model data like [CMEMS]
- HYDROLIB-core for reading polyline (red line)
- xarray for reading netcdf data and .interp() to boundary support points (red line)
- HYDROLIB-core for writing of bc/ext files
- Examples on first support point: waterlevel, NO3, ux/uy





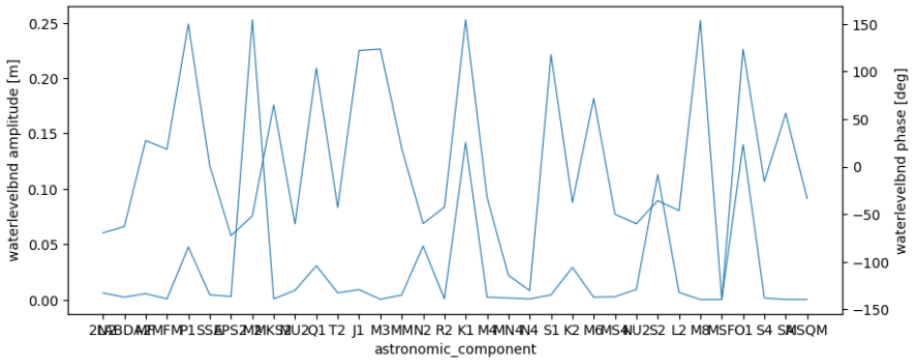


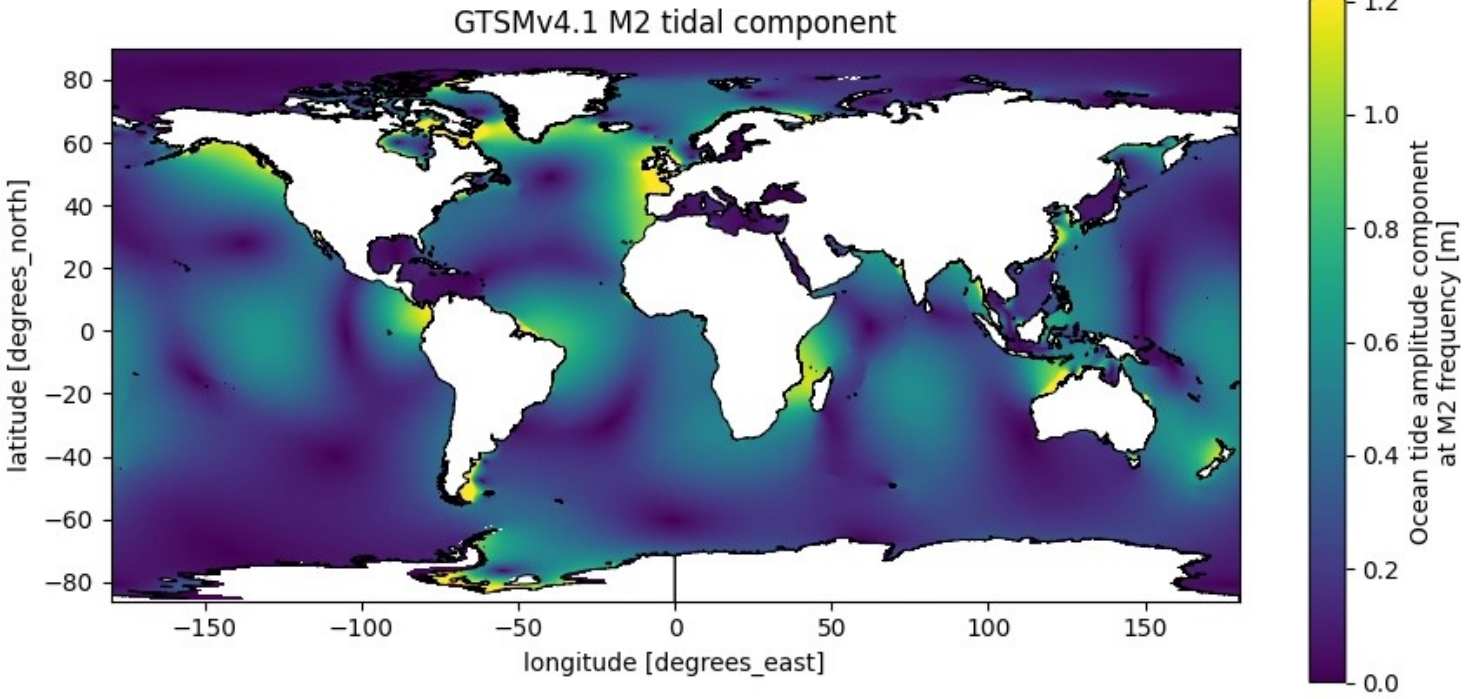




Interpolation of tidal models to boundary support points (.bc files)

- Tidal waterlevel boundary
- Interpolate tidal components to points in polyfile (with xarray)
- Available sources: FES2014, FES2012, EOT20, GTSMv4.1
- Written to Astronomic bc files (with HYDROLIB-core)

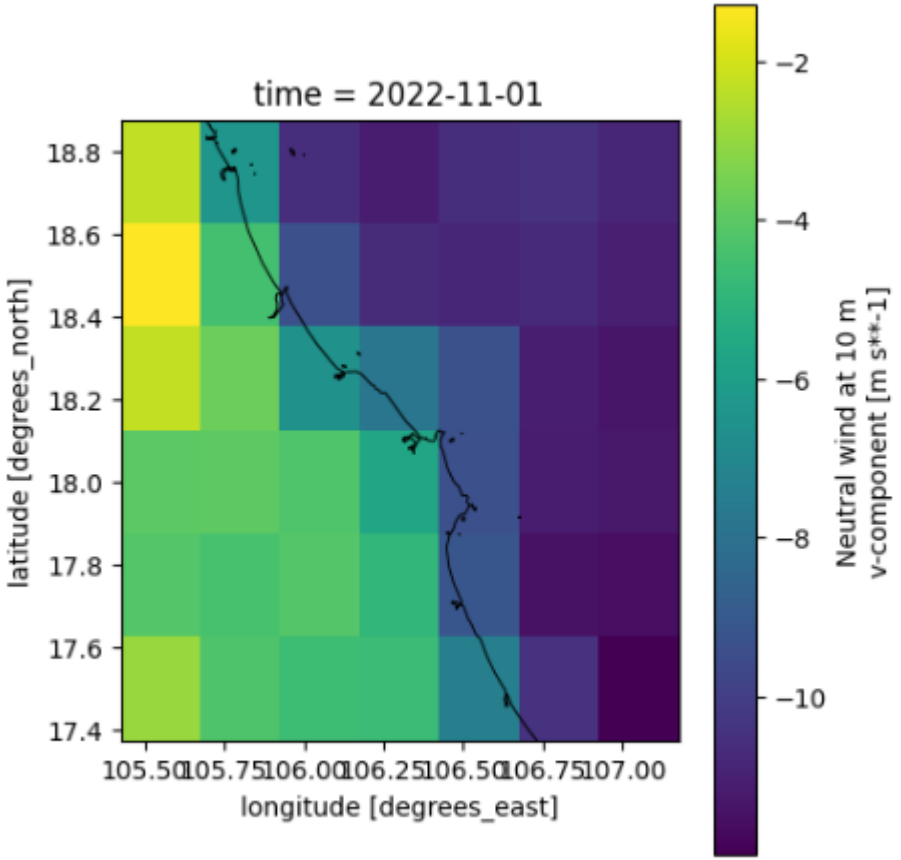




Deltares

Download and convert meteo forcing

- Download meteo data like ERA5 with cdsapi
- Merge and convert to D-FlowFM format with xarray



I/O of D-FlowFM input files

- HYDROLIB-core for reading/writing of D-FlowFM input files (.mdu, .ext, .bc, .pli, etc.)
- Validation of model structure and file contents
- Adjusting model settings (mdu)
- Full list of functionalities: https://deltares.github.io/HYDROLIB-core/latest/topics/dhydro_support/

```
# initialize mdu file and update settings
mdu_file = os.path.join(dir_output, f'{model_name}.mdu')
mdu = hcdfm.FMModel()

# add the grid (_net.nc, network file)
mdu.geometry.netfile = netfile

# add the external forcing files (.ext)
mdu.external_forcing.extforcefile = ext_file_old
mdu.external_forcing.extforcefilenew = ext_new

# update output settings
mdu.output.obsfile = file_obs
mdu.output.hisinterval = [60]
mdu.output.mapinterval = [1800]#[86400]
mdu.output.rstinterval = [0]
mdu.output.statsinterval = [3600]

# save .mdu file
mdu.save(mdu_file) # ,path_style=path_style)

# visualize the model tree
mdu.show_tree()
```

```
Vietnam.mdu
  Geometry
   └ Vietnam_net.nc
  ExternalForcing
   └ Vietnam_old.ext
   └ Vietnam_new.ext
     └ Boundary
       └ Vietnam.pli
       └ tide_Vietnam_tpxo80_opendap.bc
     └ Boundary
       └ Vietnam.pli
       └ waterlevelbnd_Vietnam_CMEMS.bc
     └ Boundary
       └ Vietnam.pli
       └ salinitybnd_Vietnam_CMEMS.bc
     └ Boundary
       └ Vietnam.pli
       └ temperaturebnd_Vietnam_CMEMS.bc
     └ Boundary
       └ Vietnam.pli
       └ uxuyadvectionvelocitybnd_Vietnam_CMEMS.bc
  Output
   └ Vietnam_obs.xyn
```

Try it yourself

- Open the modelbuilder_example.ipynb from the dfm_tools Github

Post-processing: visualization of model output

I/O of netcdf files

- lazy loading of netcdf files with knowledge of the entire structure
- Dask for chunking, parallelization
- easy indexing over all dimensions with .isel() and .sel()
- preprocess_his() enables indexing via station names (and other labels)

```
import xarray as xr
import matplotlib.pyplot as plt
plt.close('all')

from dfm_tools.xarray_helpers import preprocess_hisnc

file_nc =  r'c:\DATA\dfm_tools_testdata\DFM_3D_z_Grevelingen\\computations\\run01\\DFM_OUTPUT_Grevelingen-FM\\Grevelingen-FM_000

data_xr = xr.open_mfdataset(file_nc, preprocess=preprocess_hisnc)
```

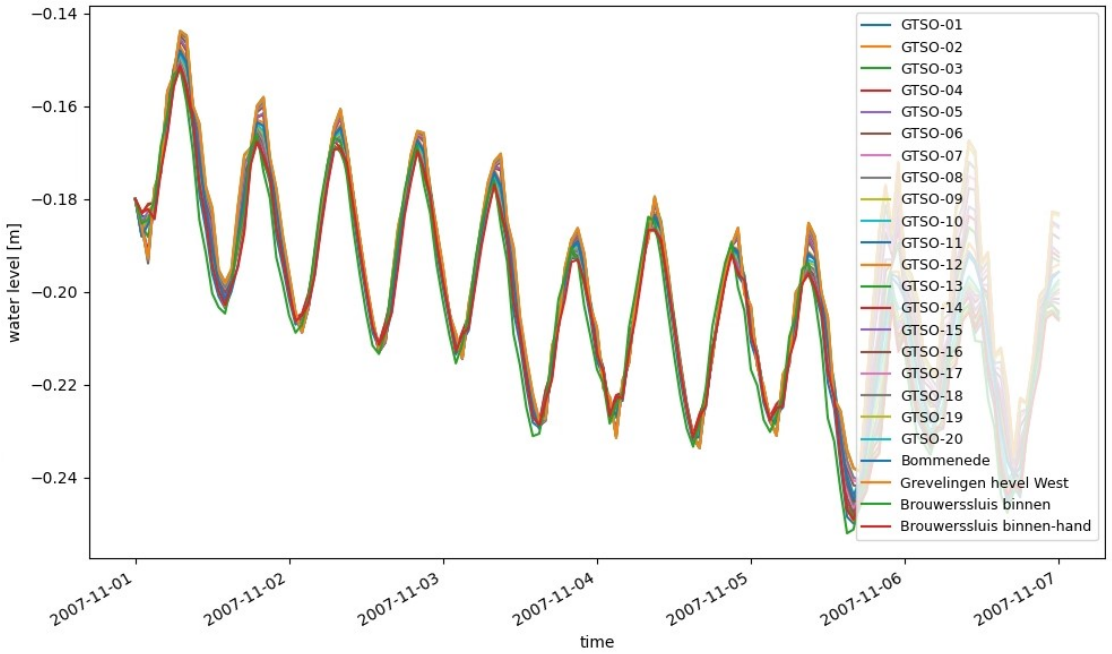
```
Out[41]:
<xarray.Dataset>
Dimensions:                                          (stations: 24, time: 145,
                                                      laydim: 36, laydimw: 37,
                                                      source_sink_pts: 2,
                                                      source_sink: 1)
Coordinates:
    station_x_coordinate                             (stations) float64 4.747e+0...
    station_y_coordinate                             (stations) float64 4.188e+0...
    zcoordinate_c                                    (time, stations, laydim) float64 dask.array<chunksize=(145, 24, 36), meta=np.n
    zcoordinate_w                                    (time, stations, laydimw) float64 dask.array<chunksize=(145, 24, 37), meta=np.
  * time                                             (time) datetime64[ns] 2007-...
  * stations                                         (stations) <U25 'GTSO-01' ....
  * source_sink                                      (source_sink) <U2 ''
Dimensions without coordinates: laydim, laydimw, source_sink_pts
Data variables: (12/42)
    station_id                                       (stations) |S64 dask.array<chunksize=(24,), meta=np.ndarray>
    waterlevel                                       (time, stations) float64 dask.array<chunksize=(145, 24), meta=np.ndarray>
    bedlevel                                         (stations) float64 dask.array<chunksize=(24,), meta=np.ndarray>
    waterdepth                                       (time, stations) float64 dask.array<chunksize=(145, 24), meta=np.ndarray>
    x_velocity                                       (time, stations, laydim) float64 dask.array<chunksize=(145, 24, 36), meta=np.n
    y_velocity                                       (time, stations, laydim) float64 dask.array<chunksize=(145, 24, 36), meta=np.n
    ...                                               ...
```

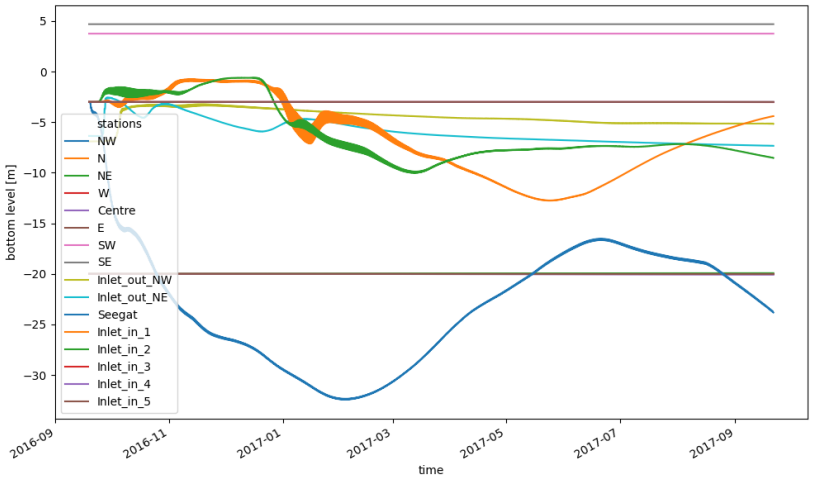
Deltares

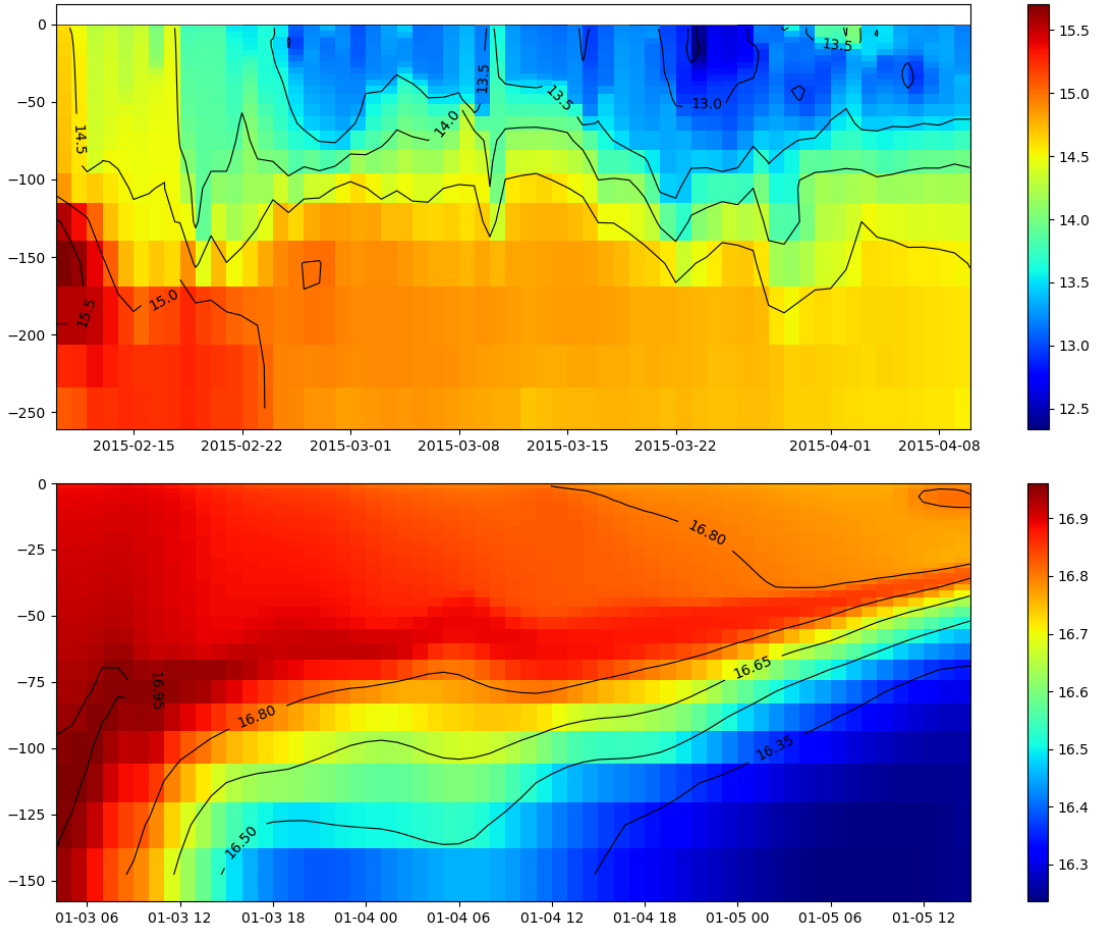
Processing hisfiles

- xarray for reading D-FlowFM hisfile
- subsetting of data over dimensions: time, depth, stations, cross_sections, structures, etc
- Easy and flexible plotting

```
data_fromhis_xr = data_xr.waterlevel.sel(stations=stations_requested,time=slice('2007-11-01','2011-11-07'))
fig, ax = plt.subplots(figsize=(10,6))
data_fromhis_xr.plot.line('-',ax=ax,x='time')
```

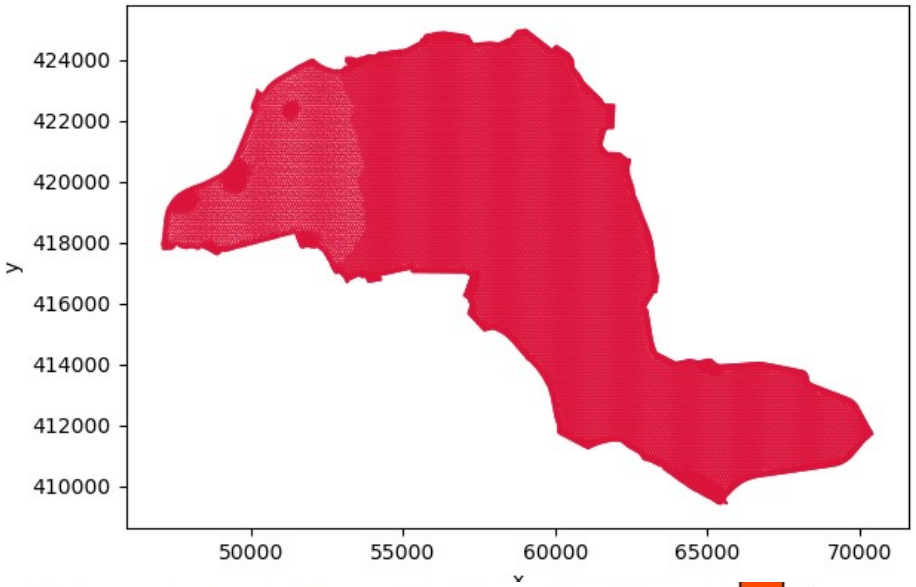


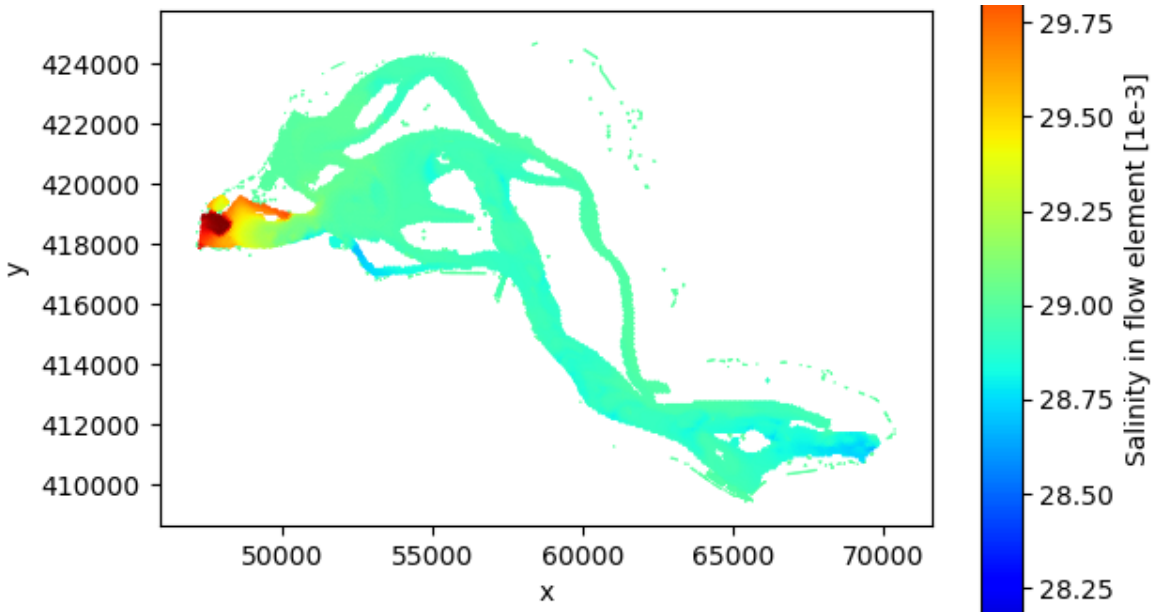


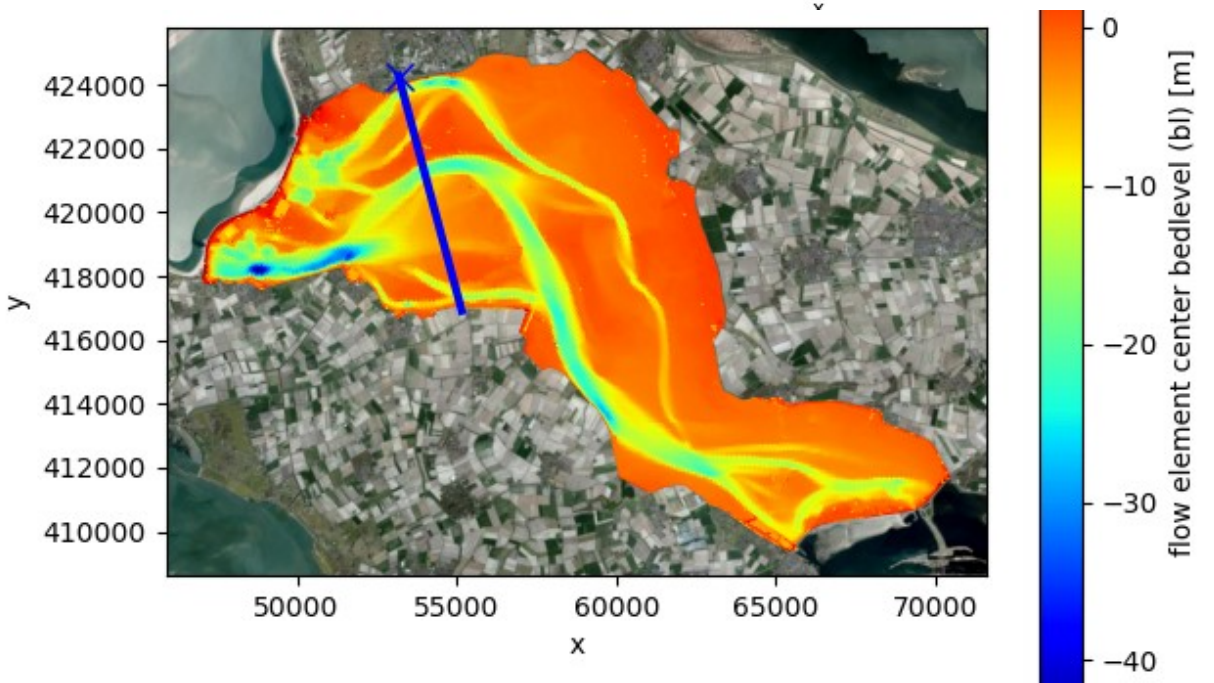


Processing mapfiles

- xarray/xugrid for reading D-FlowFM (unstructured) map/fou/rst/net files
- supports partitions (handling of ghost cells)
- select data based on variable and dimensions (time/layer/etc)
- satellite backgrounds with [contextily]

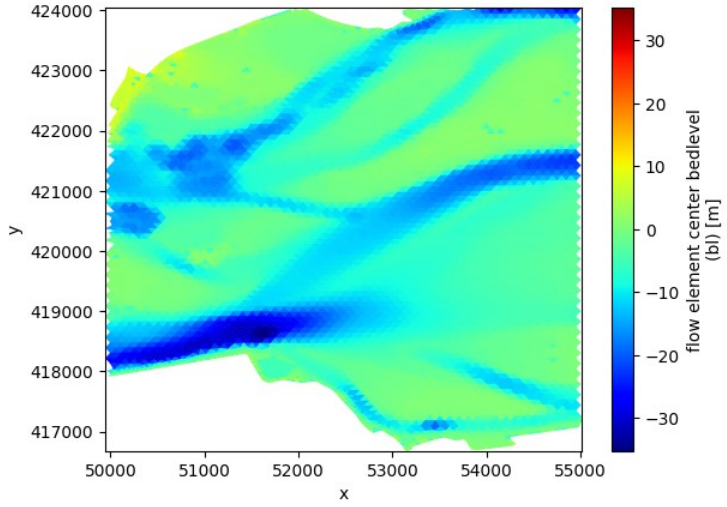


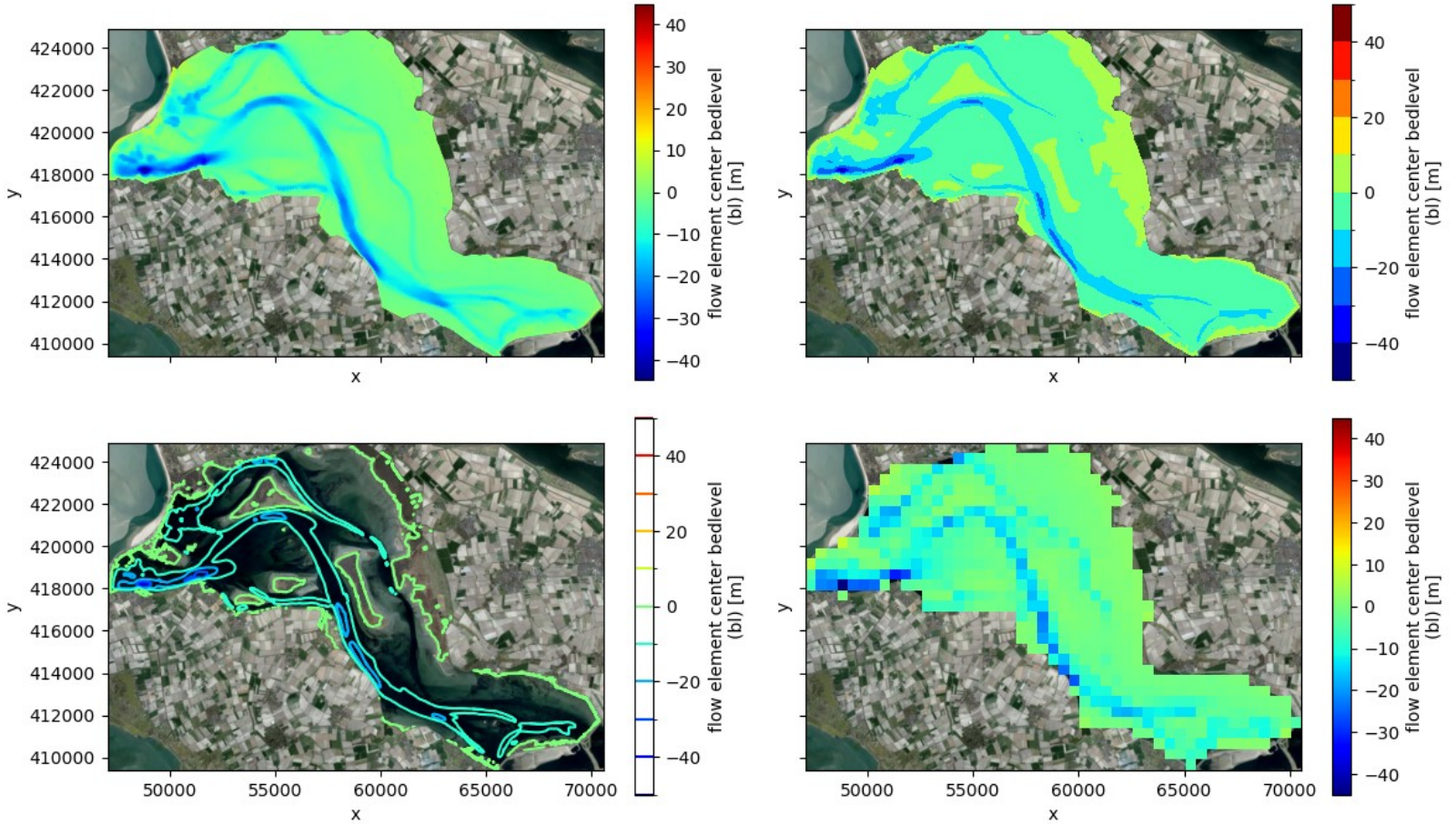




Many plotting/subsetting options

- Easy to create fancier plots like contourf or contour
- Subsetting on topology
- More examples in the [xugrid docs]

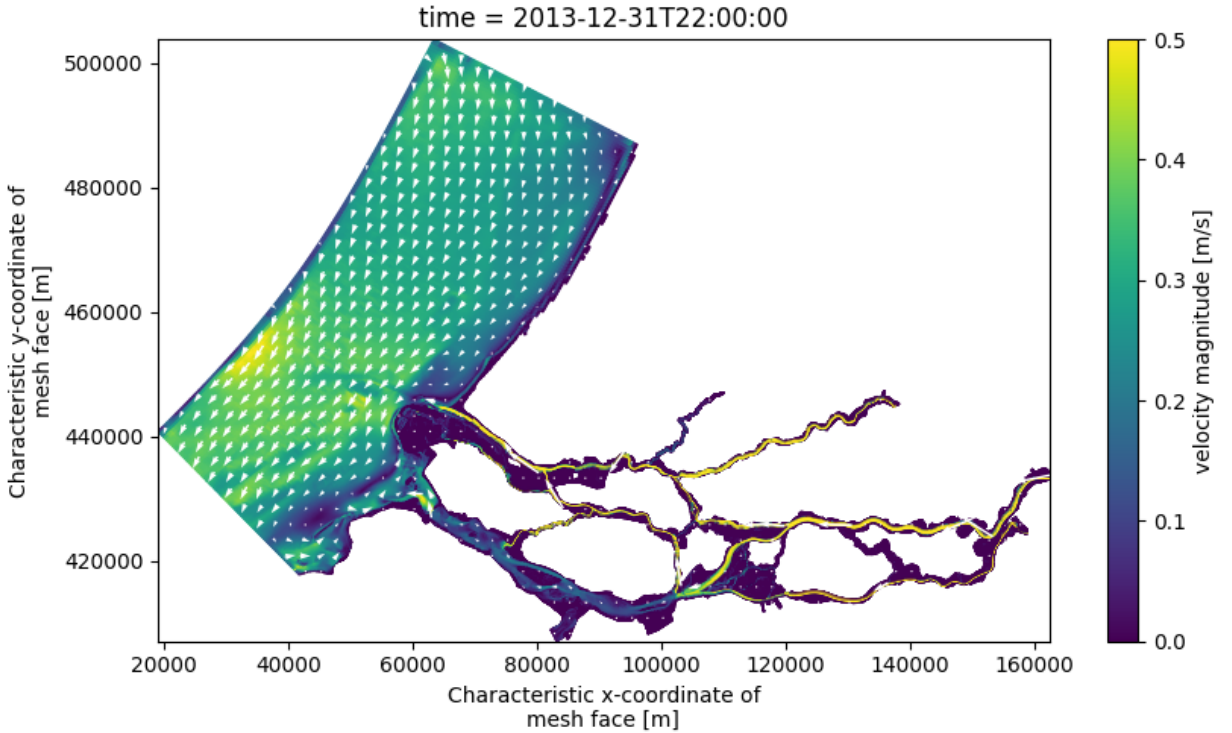


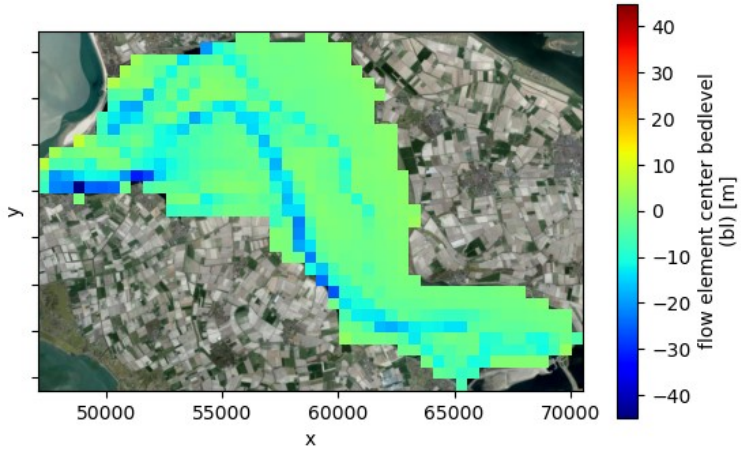


Deltares

Rasterizing and regridding

- Xugrid is a powerful tool to handle topology-related operations
- Rasterization to regular gridded dataset for easy distribution and comfortable quivers
- Regridding, reindex_like to compare model results on different grids
- More examples in the [xugrid docs]

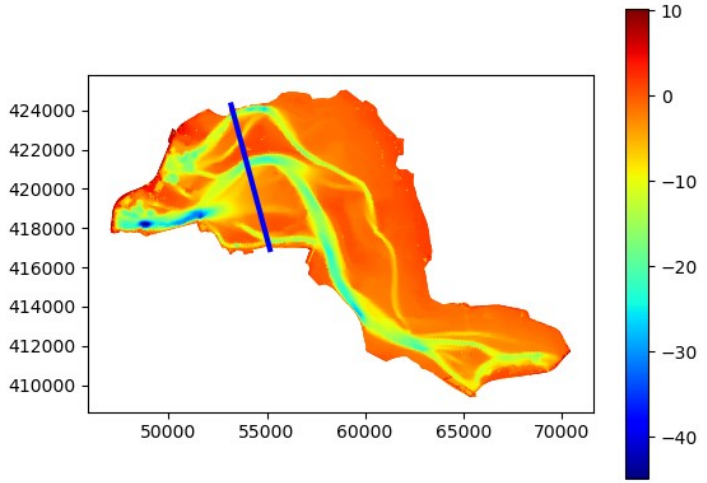


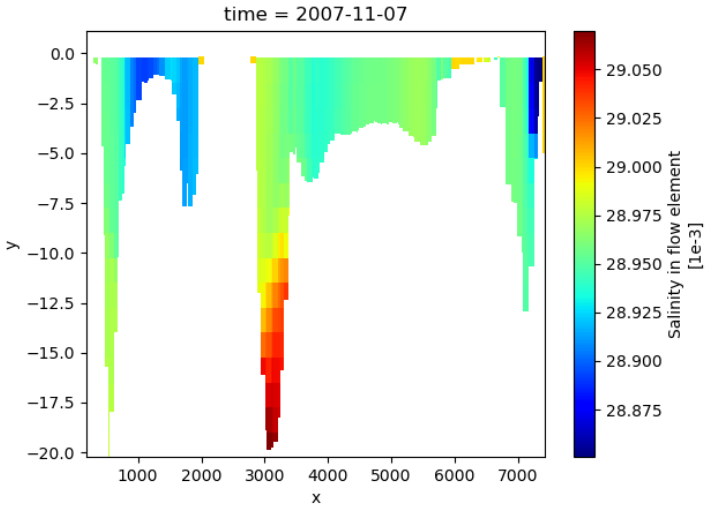


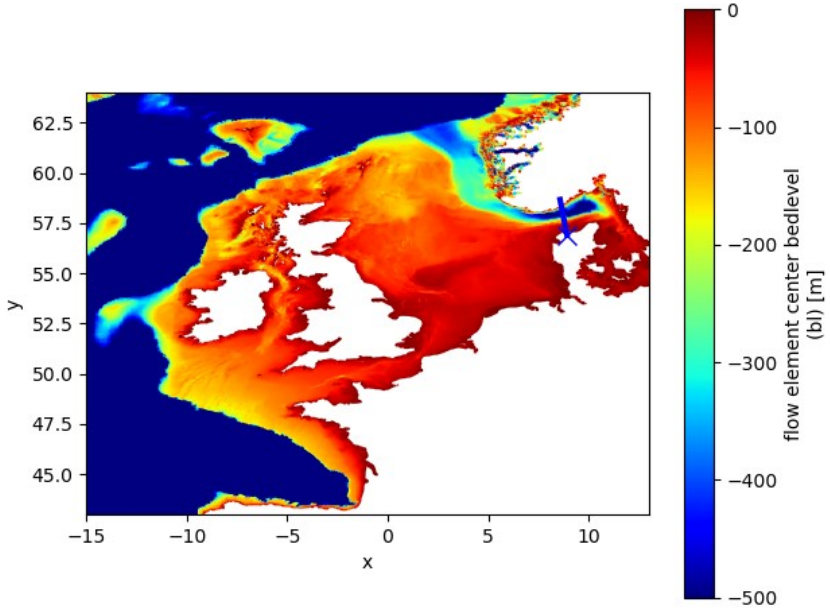
Deltares

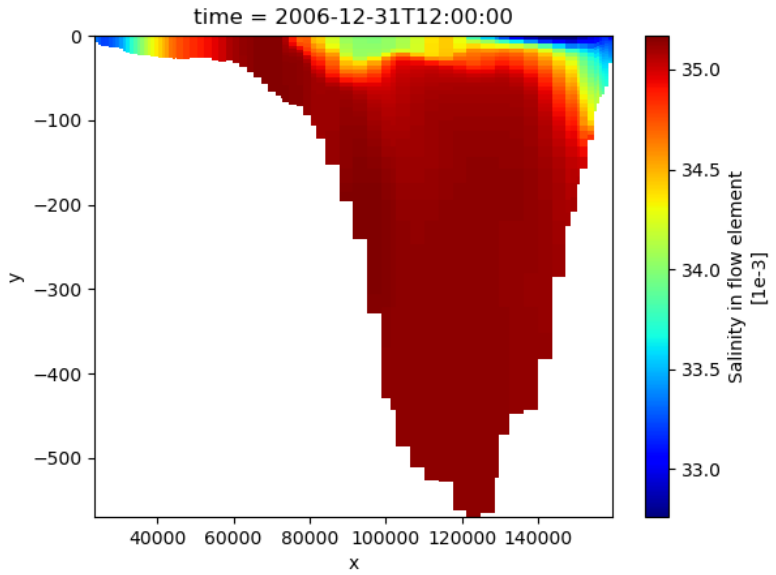
Slicing trough 3D partitioned FM map data

- Along a cross-section
- given a polyline (blue)
- slice trough 3D partitioned FM map data





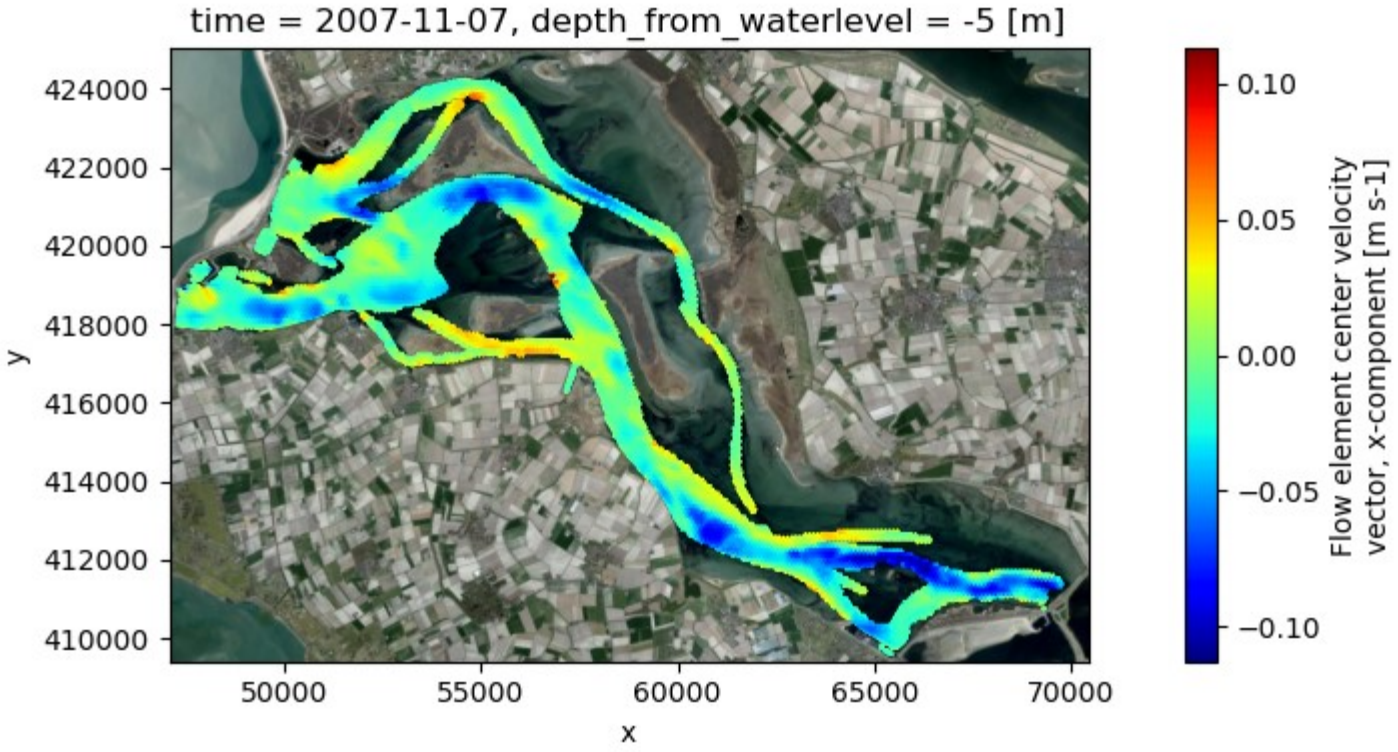




Deltares

Slicing trough 3D partitioned FM map data

- Along a depth referenced to z0, waterlevel, bedlevel
- slice trough 3D partitioned FM map data
- support for sigma, z and zsigma layers

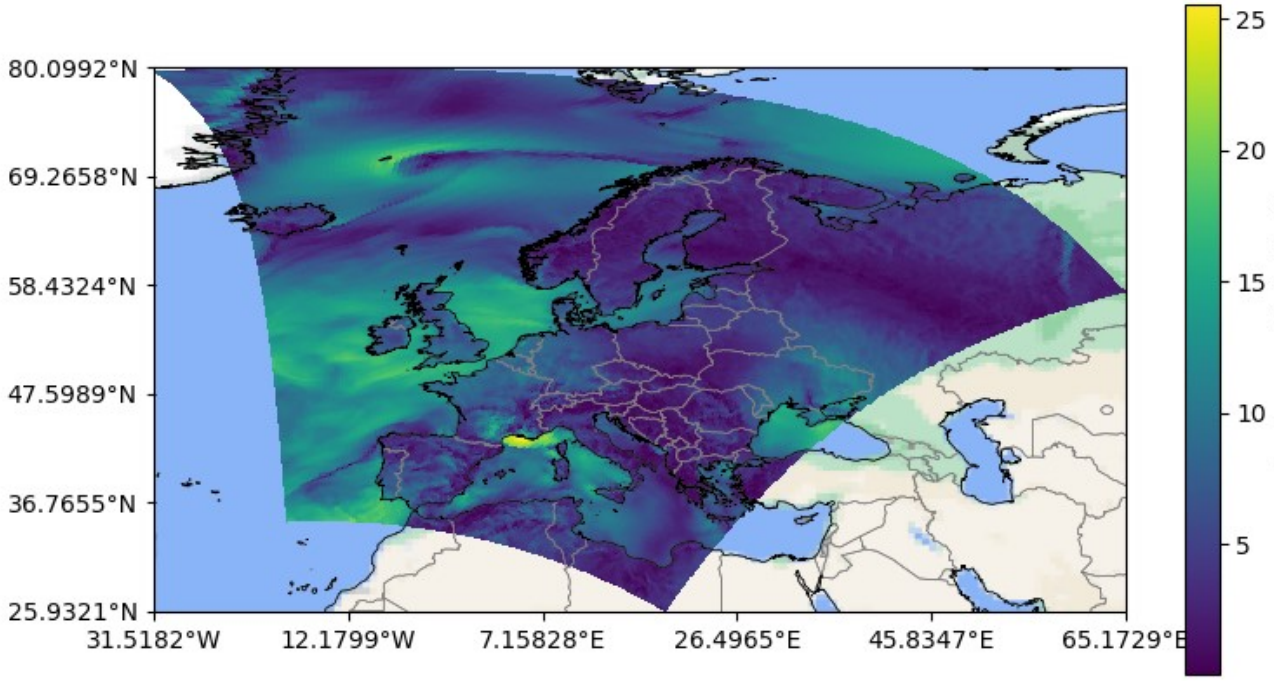


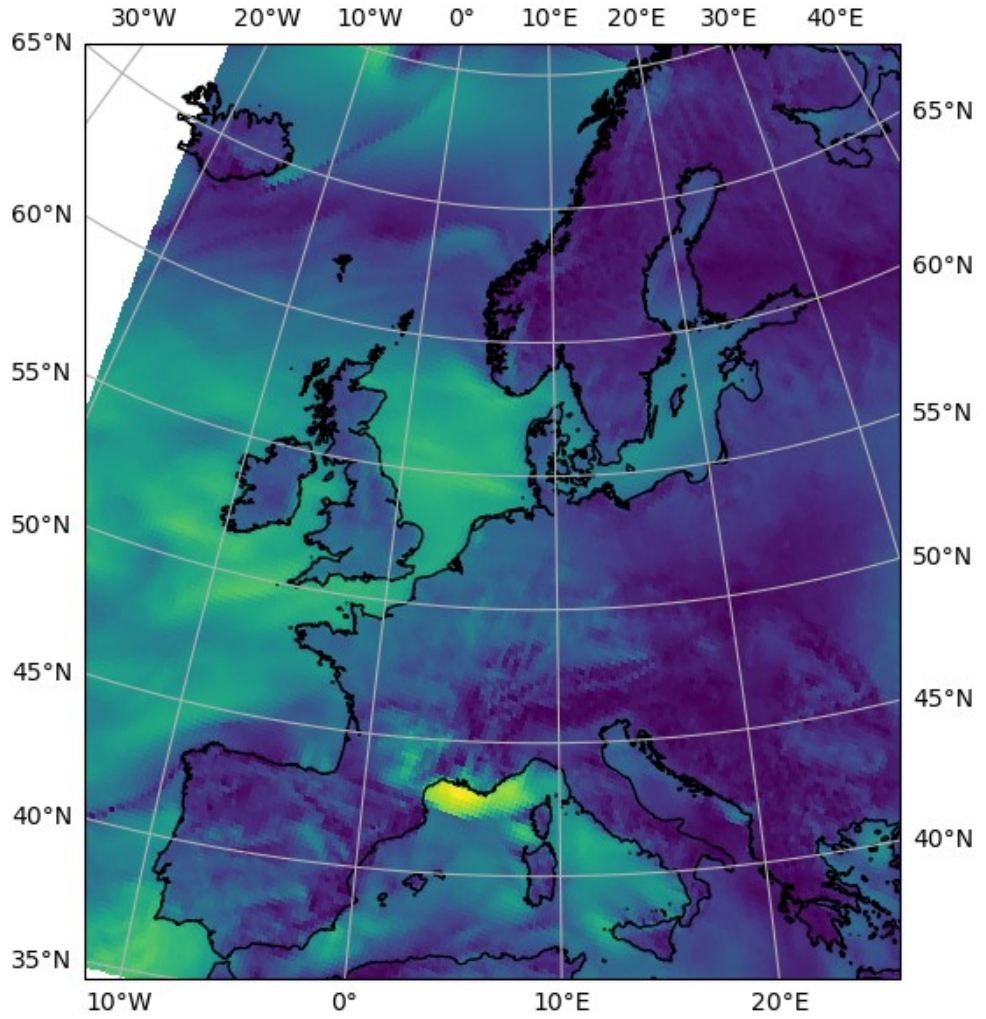
Try it yourself

- Open the postprocessing_example.ipynb from the dfm_tools Github
- More examples in the xugrid docs

Cartopy basemaps and coastlines/borders

- support for almost any epsg





Vector plots for regulargrids

- support for any other netCDF
- e.g.: SFINCS, Delft3D4, oceanmodels, meteomodels