Choptank Tributary Model

Jian Zhao[1]; Jiabi Du[2]; William Nardin[1]; Elizabeth North[1]; Lawrence Sanford[1]; Jeremy Testa[1]; Richard Tian[3]

1, University of Maryland Center for Environmental Science

2, Texas A&M University at Galveston

3, Chesapeake Bay Program Office

Major progress

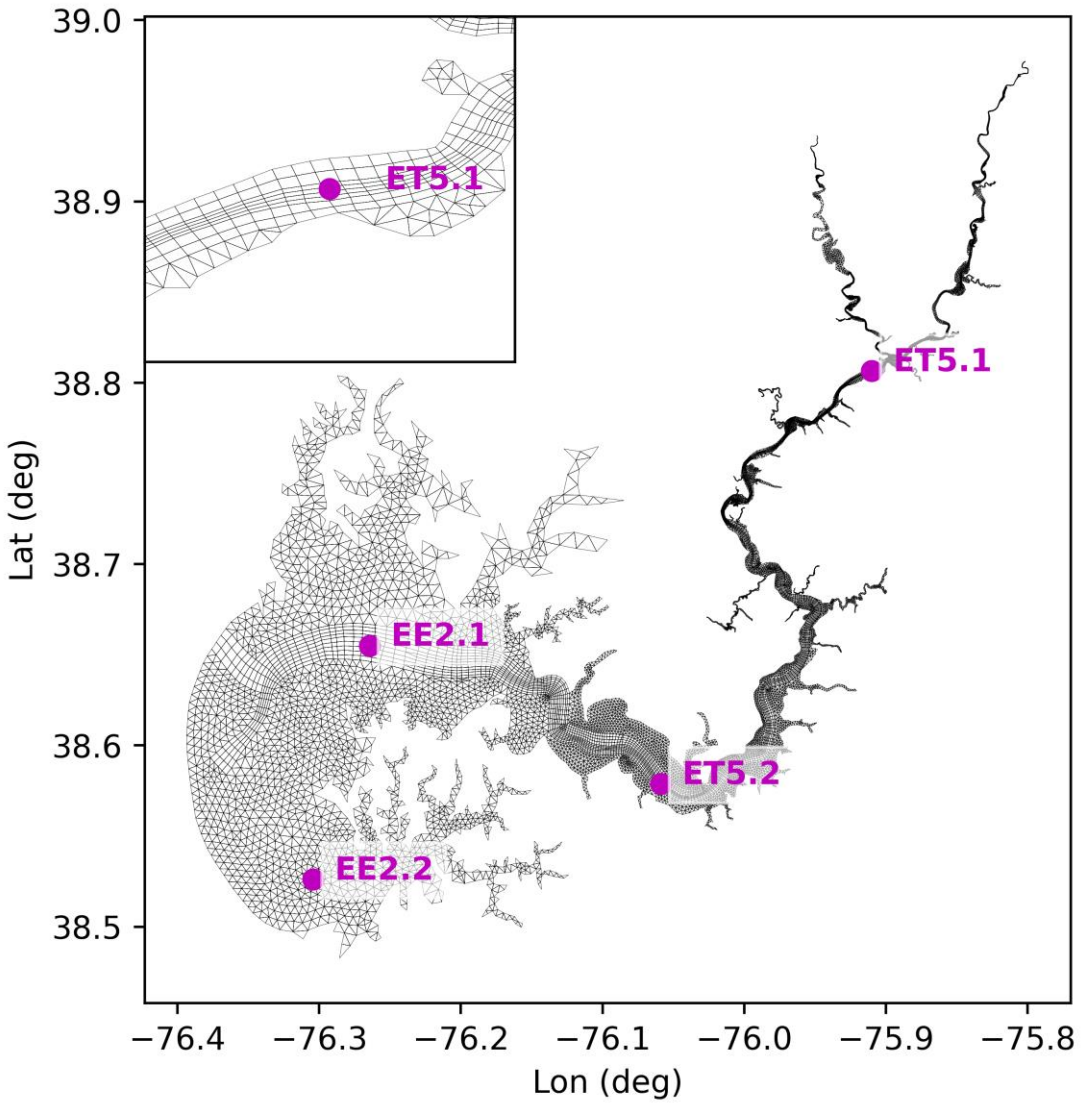
1. Hydro modeling

2. Coupled ICM

3. Standalone ICM

4. Dashboard application for model performance assessment

1. Hydrodynamic modeling



Model setup:

- Watershed Phase 6
- Open boundary condition from MBM simulation RUN07b

Figure: model grid (V1.6) and three CBP stations. Zoom-in view of grid near ET5.1 is shown in the upper left.

1. Hydrodynamic modeling

Key takeaways:

1. Using smaller drag in upper Choptank lead to stronger tide and salt intrusion

2. Flow condition (using USGS or watershed P6) also matters (not shown here)

Wenfan's test

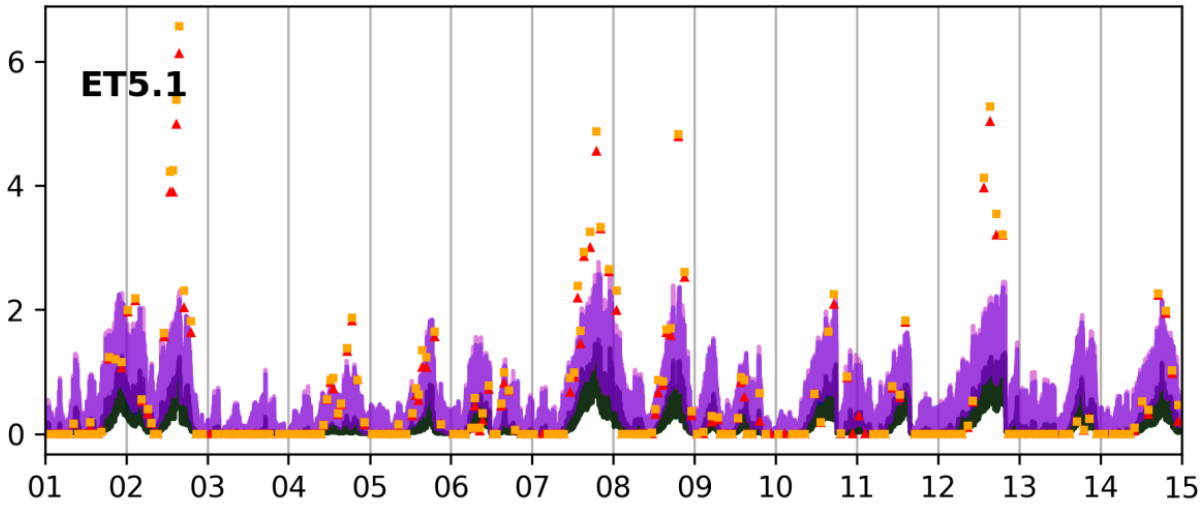


Figure: salinity at ET 5.1 from two different model run

RUN04a:base

RUN04c: drag=0 in upper Choptank

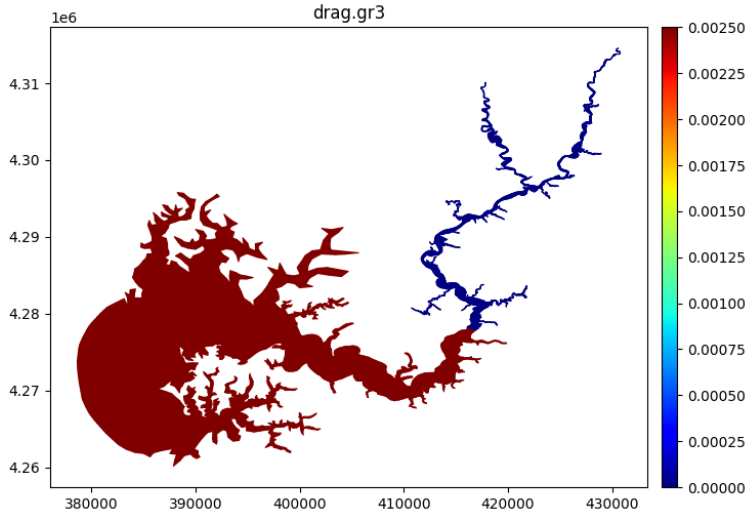


Figure: drag coefficient in RUN04c

2. Coupled ICM: setup

Using

- Watershed phase 6
- ICM boundary from mainbain RUN09k

We developed a **python package** to prepare the inputs

- Can be run in parallel mode
- Parallel mode is needed as reading ICM output from main bay model takes time
- With parallel mode and using 20 cpus, it takes ~20 minutes.

To other tributary teams
The script is available at
/sciclone/data10/jiabi/Choptank/Inputs/run16hi/add_ICM.py

```
p.StartT=datenum(1991,1,1);  p.EndT=datenum(1995,12,31) #simulation time;

p.base=None
p.grid_dir='/sciclone/data10/wangzg/MTM/grid/v0' #directory of hgrid & vg
p.grid_dir='/sciclone/data10/jiabi/Choptank/Grids/V1.6'
p.hydro_dir='../run16a' #need the hostart.nc from this folder
p.flag['ICM']= 1  #ICM model (1: 21 variables; 10: 21-variable offline mo
p.flag['SED']= 0  #SED3D model # not yet implemented
p.flag['WWM']= 0  #Wave model  # not yet implemented

p.flag['bctides.in']          = 1 #hydro &ICM
p.flag['hotstart.nc']         = 1 #hydro &ICM
p.flag['source.nc']           = 1 #hydro &ICM
p.flag['ICM_3D.th.nc']        = 1 #ICM
p.flag['ICM_nu.nc']           = 1 #ICM
p.flag['ICM_param.nc']        = 1 #ICM
p.flag['ICM_sflux.th.nc']     = 1 #ICM
p.flag['icm.nml']             = 1 #ICM
p.flag['ICM_nudge.gr3']       = 1 #ICM
p.flag['istation.in']         = 1 #ICM

#databases
p.bdir='/sciclone/data10/wangzg/MTM/database/'          #dir of setup file
p.MBM       = p.bdir+'hydro_MBM/RUN07b/'                #Main Bay Model hy
p.MBM_ICM   = '/sciclone/data10/wangzg/CBP/RUN09k/'     #ICM model run, 6

#databases
p.bdir='/sciclone/data10/wangzg/MTM/database/'          #dir of setup file
p.bdir2='/sciclone/data10/wangzg/CBP/setup_files/'
p.MBM       = p.bdir+'hydro_MBM/RUN07b/'                #Main Bay Model hy
p.MBM_ICM   = '/sciclone/data10/wangzg/CBP/RUN09k/'     #ICM model run, 6
p.sflux     = p.bdir+'sflux_narr_subdomain'             #sflux database
p.source    = p.bdir+'load_p6_v3.npz'                   #CBP watershed sou
p.region    = p.bdir2+'region/'                          #regions files
p.atmdep    = p.bdir2+'atm_load.npz'
p.station   = p.bdir2+'bp/station_CB.bp'                 #station informat
p.CBSEG=['RPPTF','RPPMH','RPPOH','CB6PH','CB5MH_VA']    #for Rappahannock
p.CBSEG=['CHOTF','CHOOH','CHOMH2','CHOMH1']              #for Choptank
```

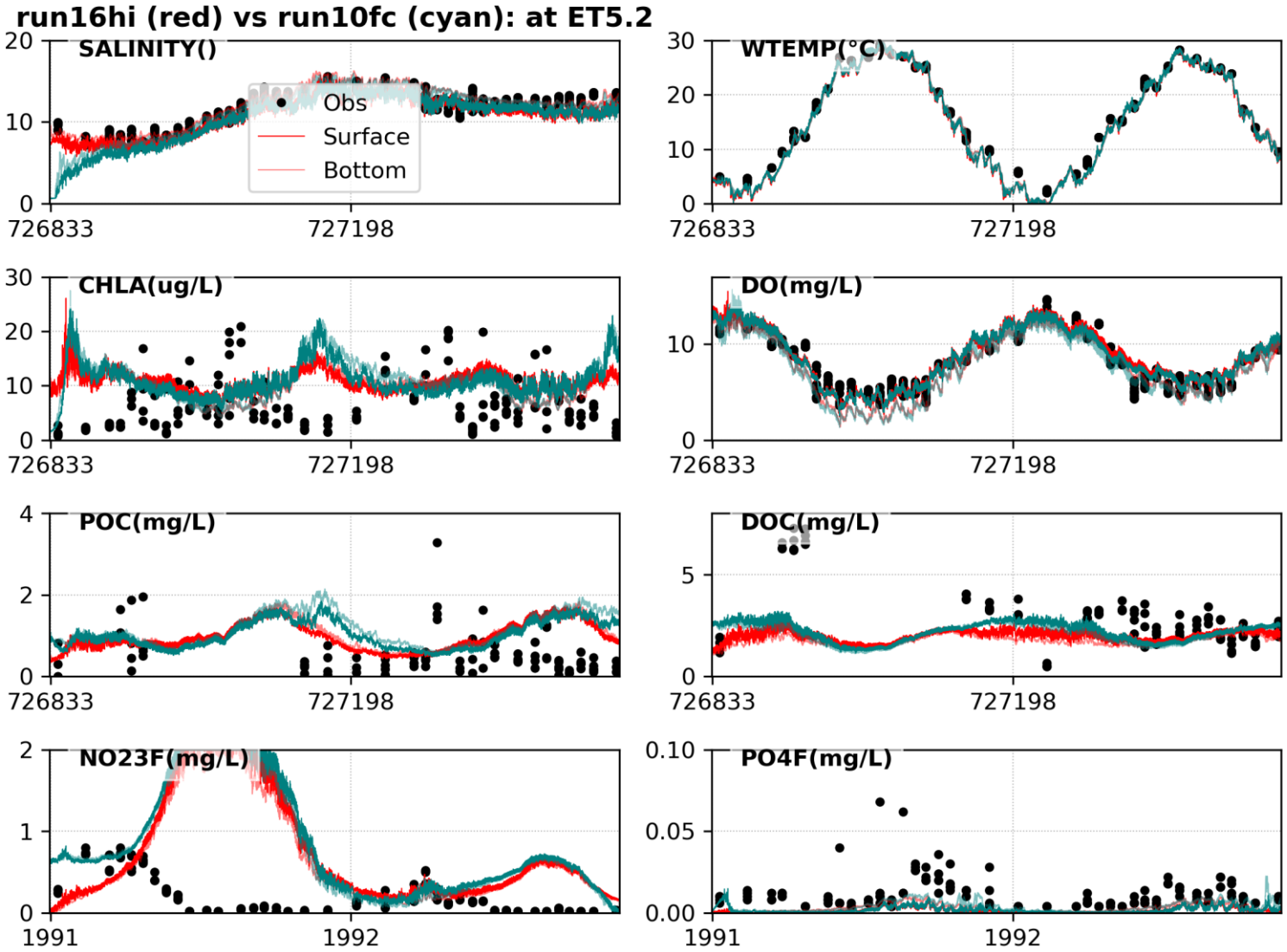
2. Coupled ICM: computational efficiency

	V1.0	V1.6
Grid size (# of nodes)	4,122	14,966
Hours for 1-year simulation with 96 cpus, 27 of which for IO	11	25

Question: Do we need higher resolution of grid?

2. Coupled ICM: compare with old grid

New grid Old grid



The difference is minimal.

It is not sensitive to the grid size and resolution.

Figure: water quality model results at ET5.2 from two model runs.

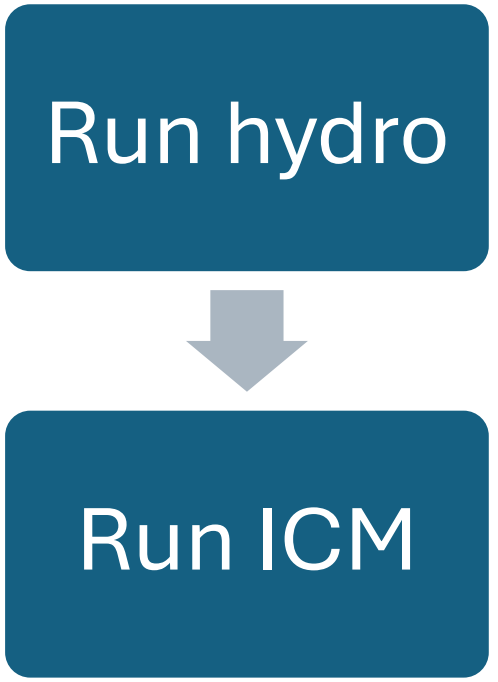
3. Standalone ICM

What if we want to use high-resolution grid and efficient ICM simulation?

Use standalone mode, in which we skip the time-consuming calculation for baroclinic process and use saved hydrodynamic output.

With same number of cpu (96), it now takes 3.5 hour for one-year simulation, **~ 8 times faster** than in coupled mode.

Steps for standalone mode



Save diffusivity, salinity, temperature, velocities

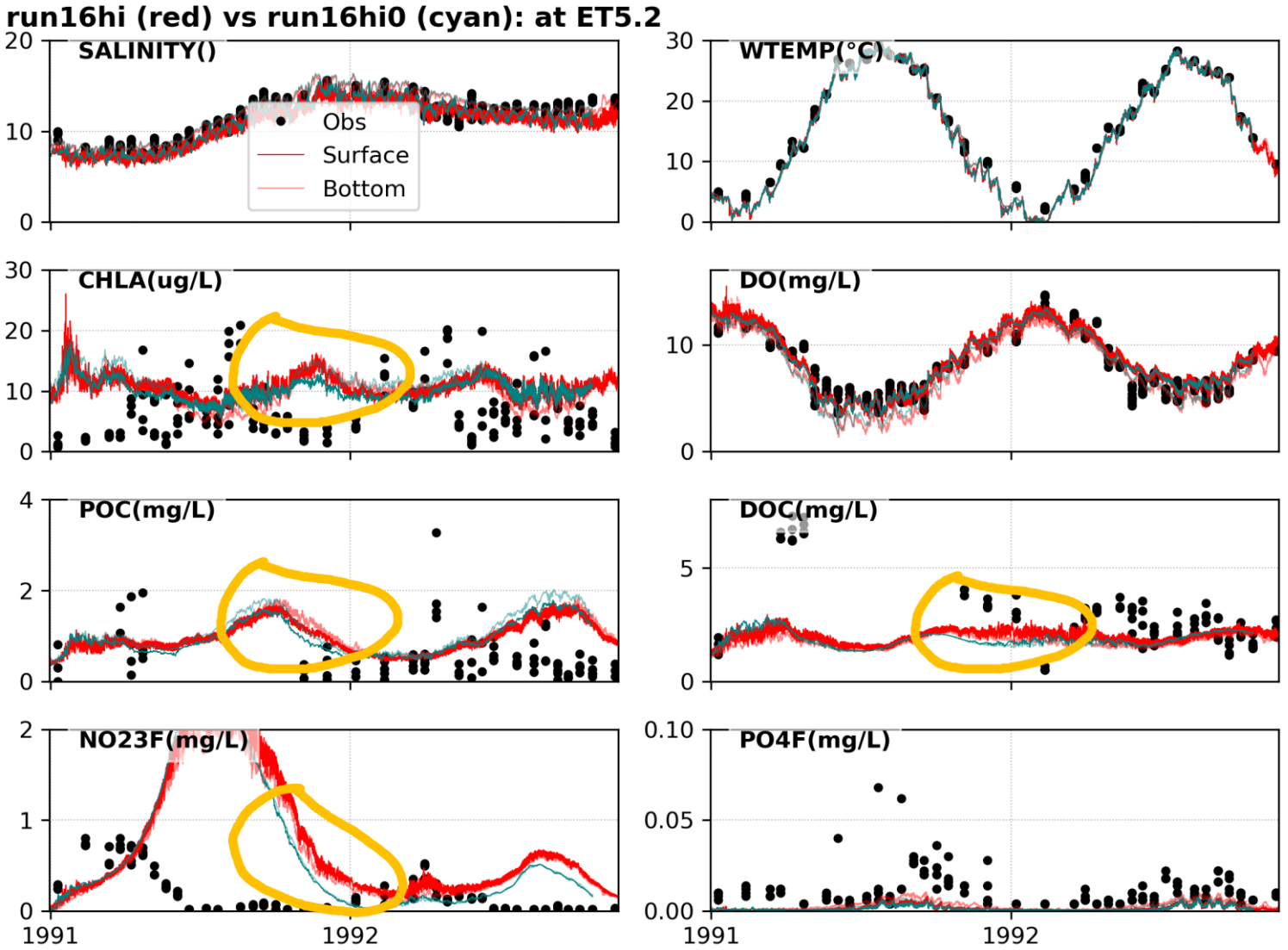
Use option itransport_only=1

In param.nml, change

- dt = 1800
- wtiminc = 1800
- nadv = 1

3. Standalone ICM: results

Coupled vs **Standalone**



The only notable difference is in the late 1991

The difference is minimal.

We are safe to using standalone mode.

3. Standalone ICM: lessor learned

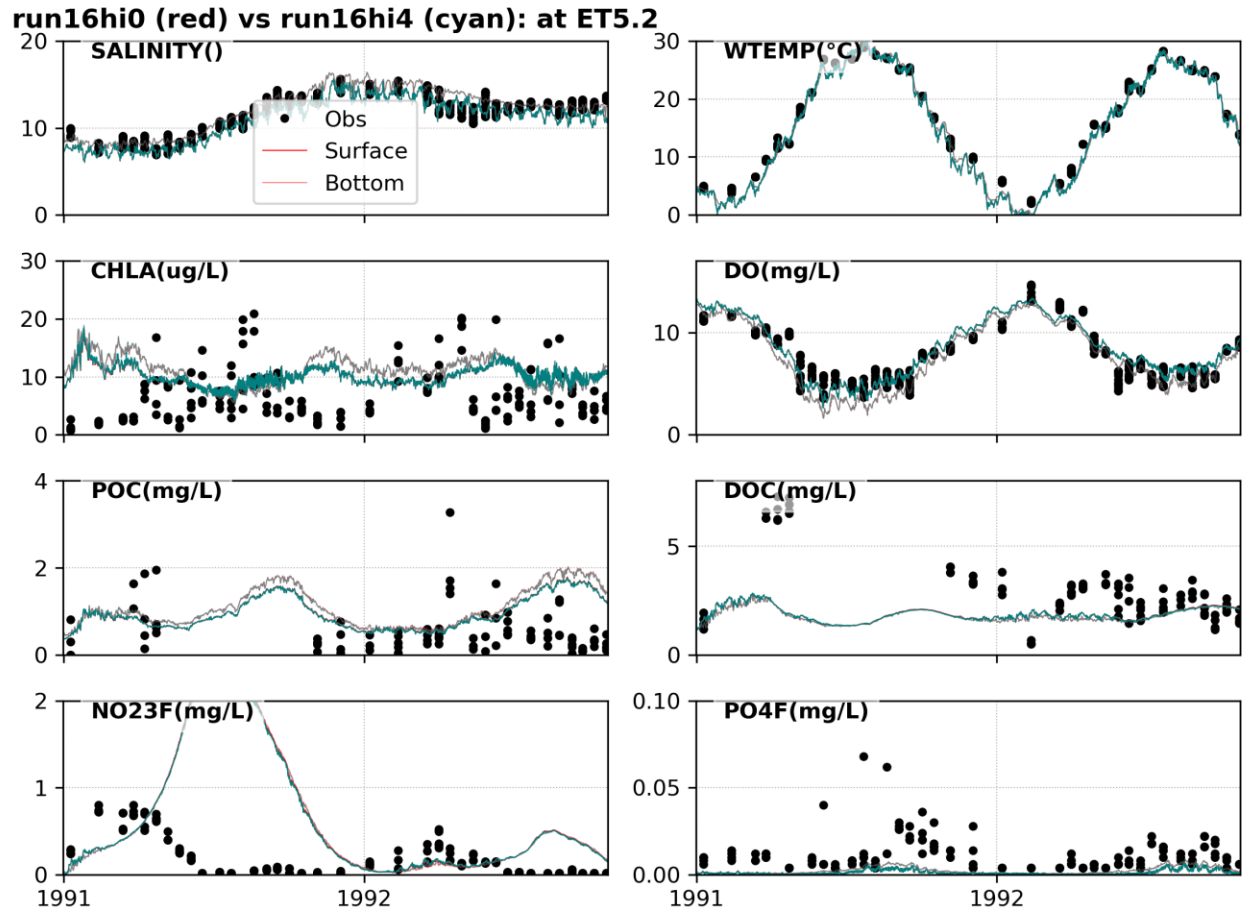
If you ends up with following error

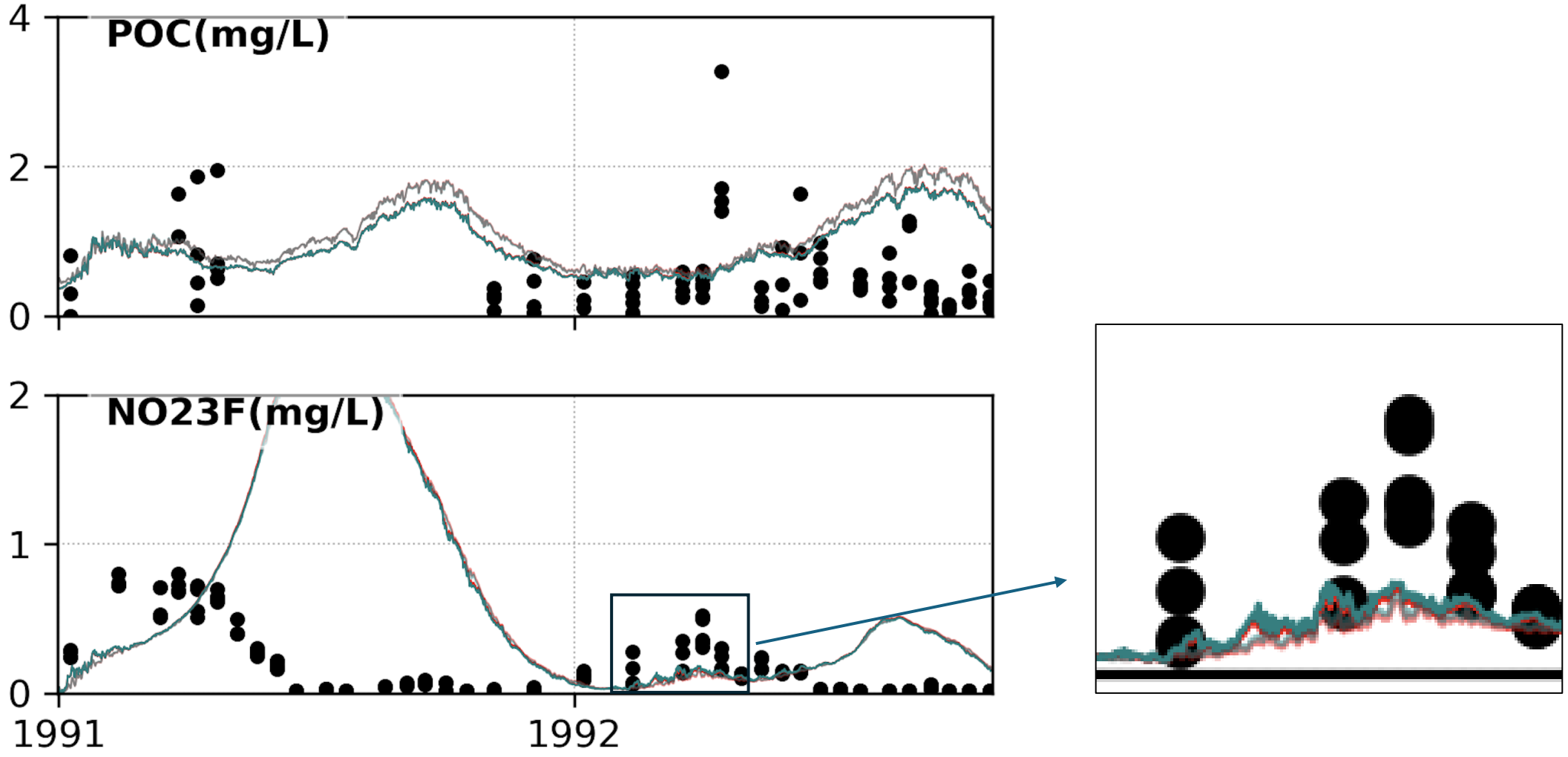
```
   3: ABORT: BTRACK: adp. wrong
Abort(0) on node 3 (rank 3 in comm 496):
```

Try using option
nadv = 1

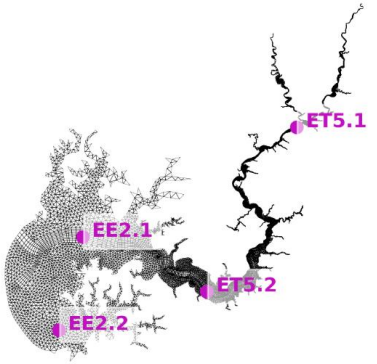
With this option, the backtracking is done using Euler method.

There is nearly no difference between two options

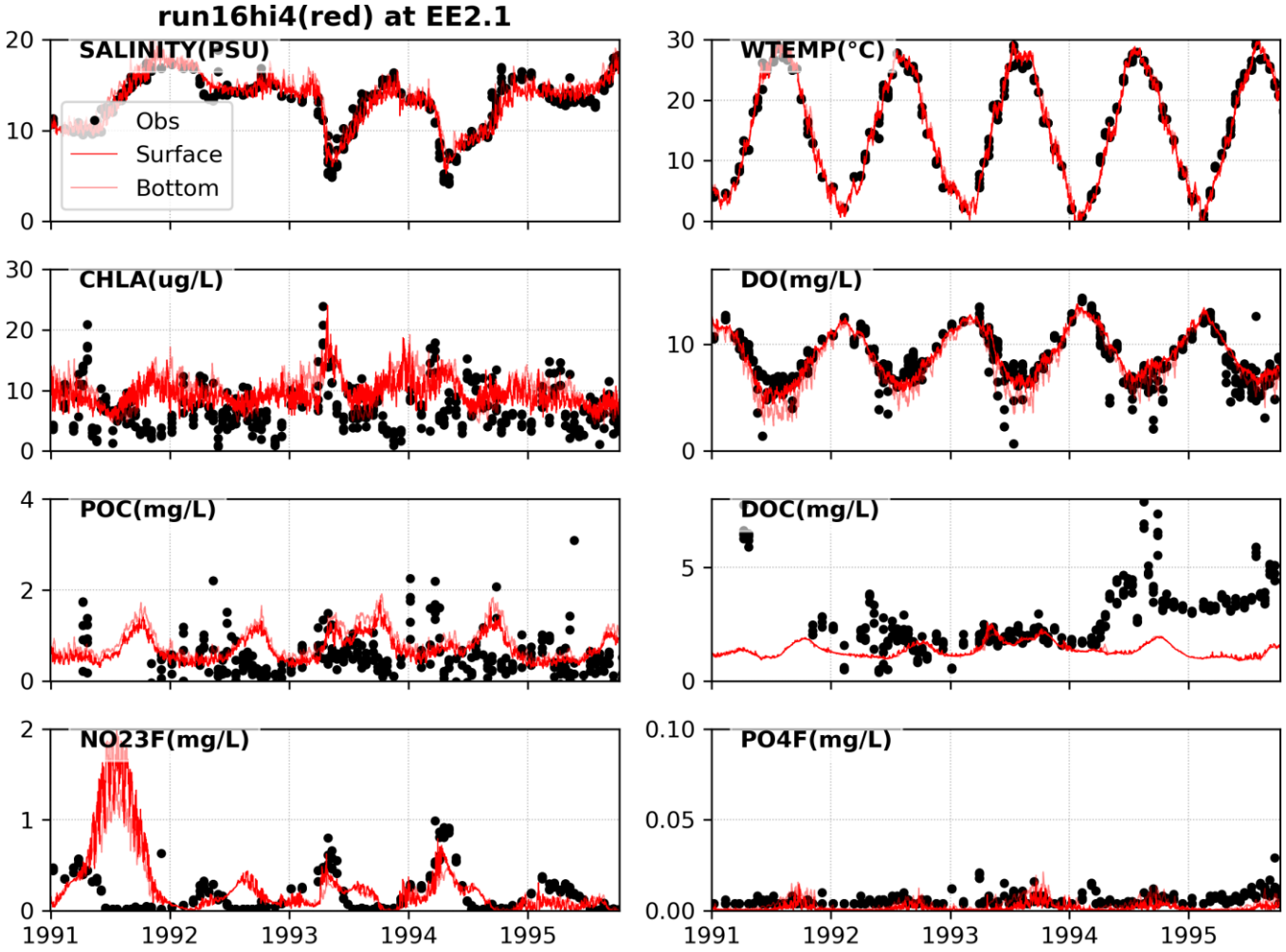



POC(mg/L)
4
2
0
NO23F(mg/L)
2
1
0
1991
1992

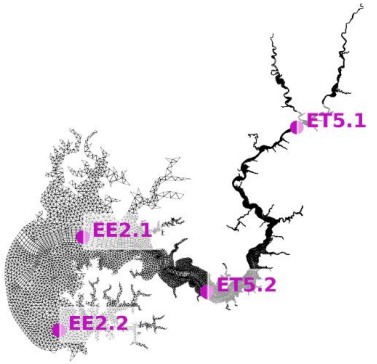
3. Standalone ICM: 5-year simulation

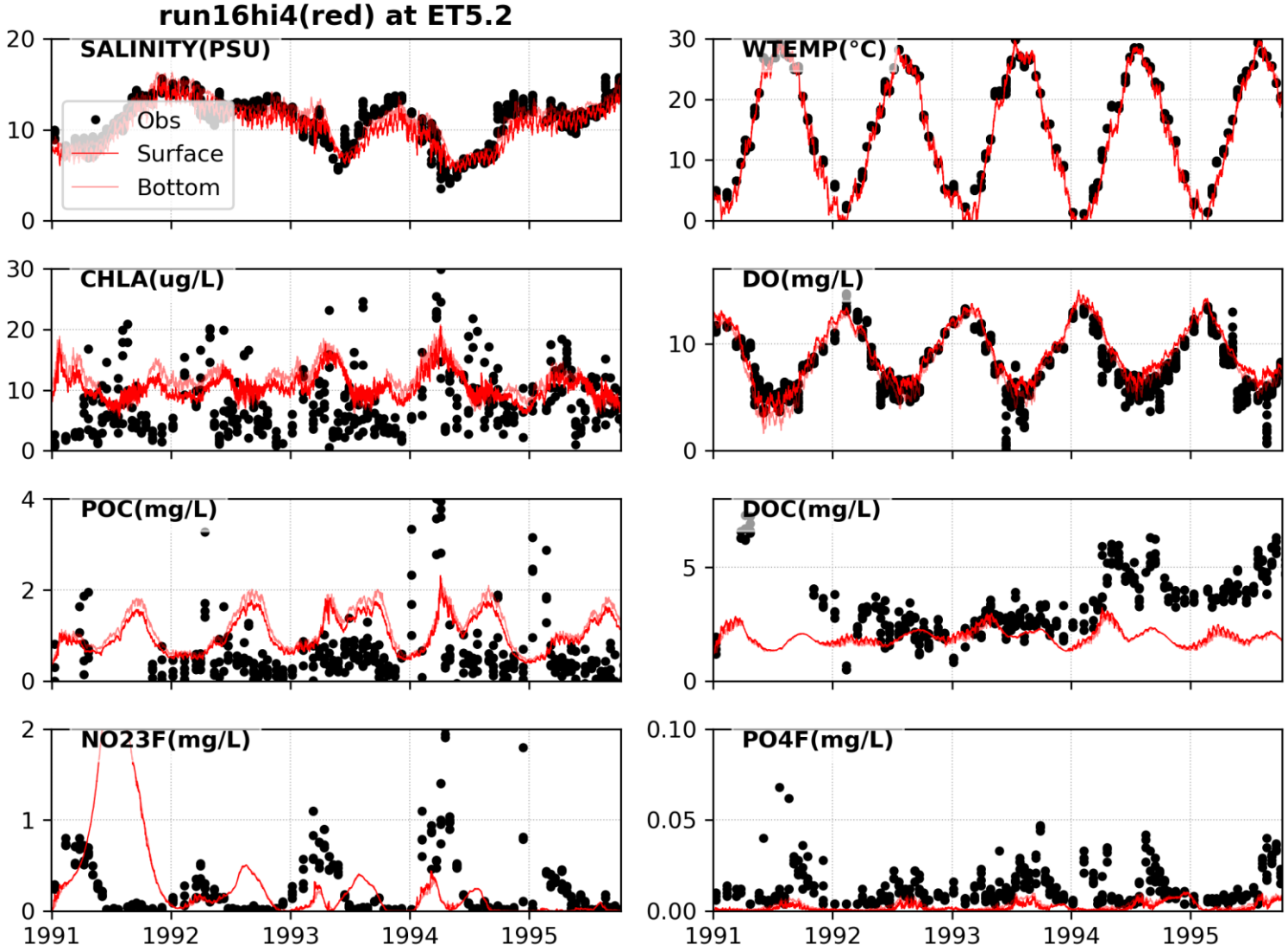


At Station EE2.1



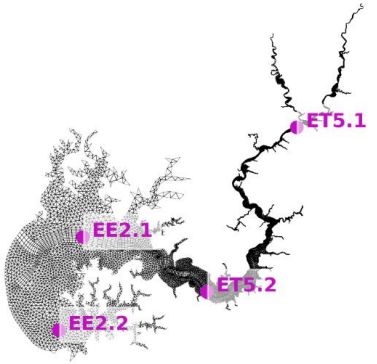
3. Standalone ICM: 5-year simulation

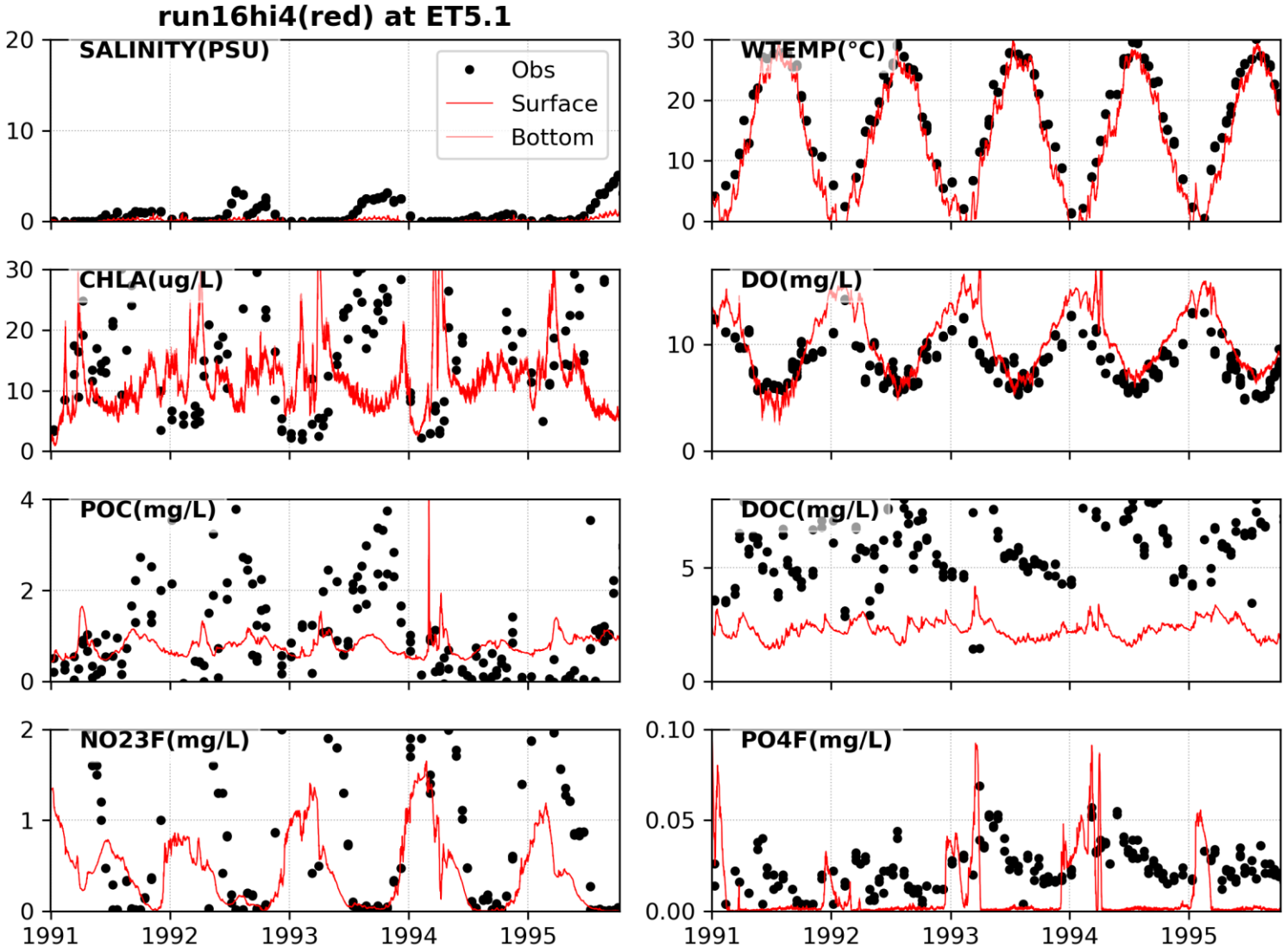




At Station ET5.2

3. Standalone ICM: 5-year simulation

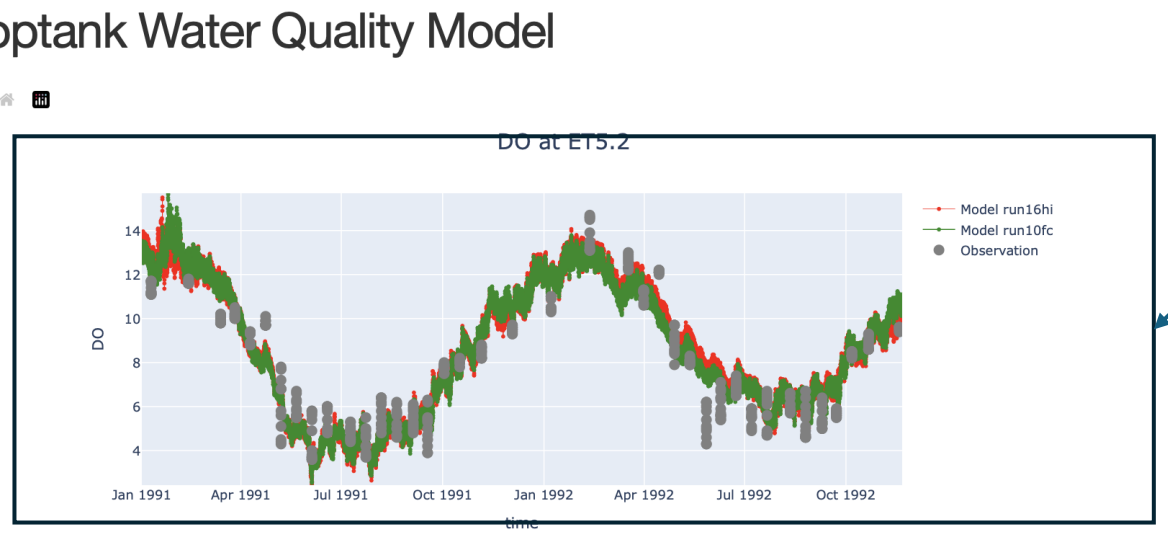




At Station ET5.1

Dashboard application for model performance assessment

1. Interactive map, where users can click the location



2. Time series plot, showing both observation and model results

3. User inputs, variable, model run, line color,

To do list:

- Add option to allow low-pass filter
- Mount on a server and make it accessible to public

Dash application available at
http://67.205.184.137:8001

Check the difference between base run (run16hi4) and better salt-intrusion run (run16ii)

Summary

There is little difference in coupled and standalone ICM simulations.

Accurately simulating salt intrusion matters.

We developed python packages to

- prepare ICM inputs for tributary model (available on sciclone)
- analyze model performance (in development; you are welcome to use)