Progress for Phase 7 Main Bay Model

VIMS team: Joseph Zhang, Jian Shen, Harry Wang, Marjy Friedrichs,

Zhengui Wang, Hyungju J. Yoo, Pierre St-Laurent

CBPO: Nicole Cai, Richard Tian

UMCES team: Jeremy Testa

January 10, 2024

Outline

- **Improve the simulation of DO and bottom temperature**
- **Add shoreline erosion and atmospheric deposition loadings**
- **Update modeling workflow**
- **Remaining issue on watershed loading**

Check on Project Tasks: 2021-2024

- We are in the beginning of 3rd year
- We are on track…

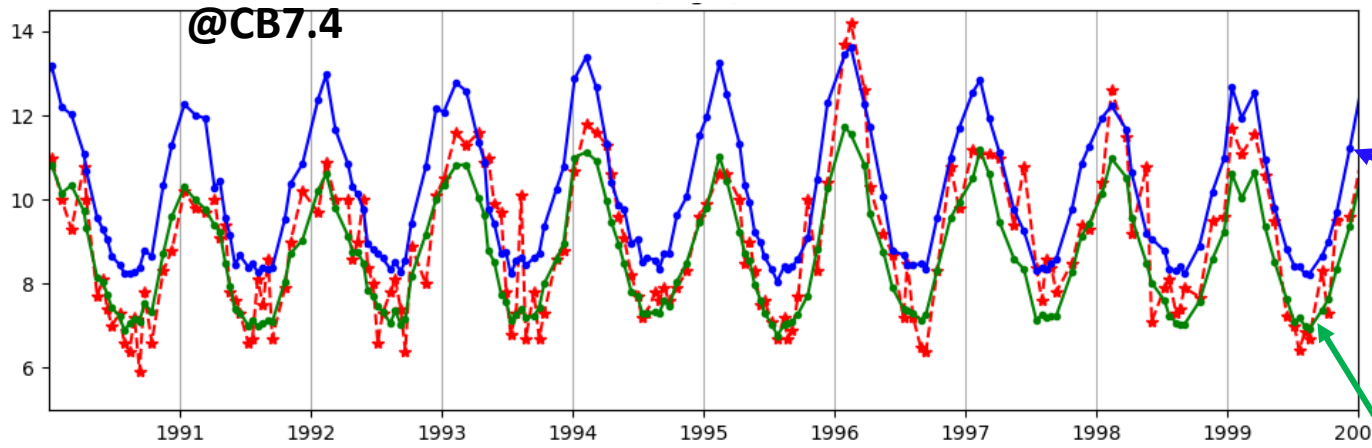
Task	Time	Description	Progress
1	Yr 1	hold an online kick-off meeting	100%
2	Yr 1	integrate the latest changes in CBP's ICM into SCHISM-ICM	99%
3	Yr 2	revise the current SCHISM Bay mesh and work on performance tuning of SCHISM-ICM	90%
4	Yr 1-2	working with watershed, airshed, hydrological modeling groups to ensure the coupling, scale, and the interface mechanisms are properly executed, including C.C. input information	80%
5	Yr 2-3	conduct full calibration and verification of hydrodynamic and WQ model output	80%
6	Yr 2-3	finish documentation on the software package	50%

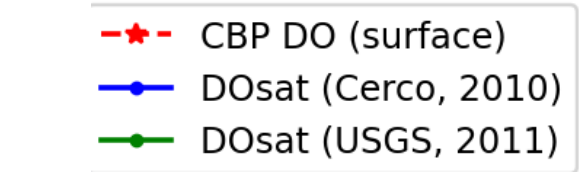
Task	Time	Description	Progress
7	Yr 3	transfer of the software package to CBPO for operational testing	20%
8	Yr 3, 6	develop user-friendly interfaces with model software and hold trainings for user support	50%
9	Yr 3	addressing some important knowledge gaps in ICM	20%
10	Yr 2-4	review all recent studies related to Bay WQ processes and work with CBP and MW to identify key missing processes, updating the code to address knowledge gaps as they are filled	50%
11	Yr 5-6	transfer the updated code version to CBPO for testing	
12	Yr 4-6	develop and apply management scenarios (including C.C.)	

Improvement #1: bottom DO

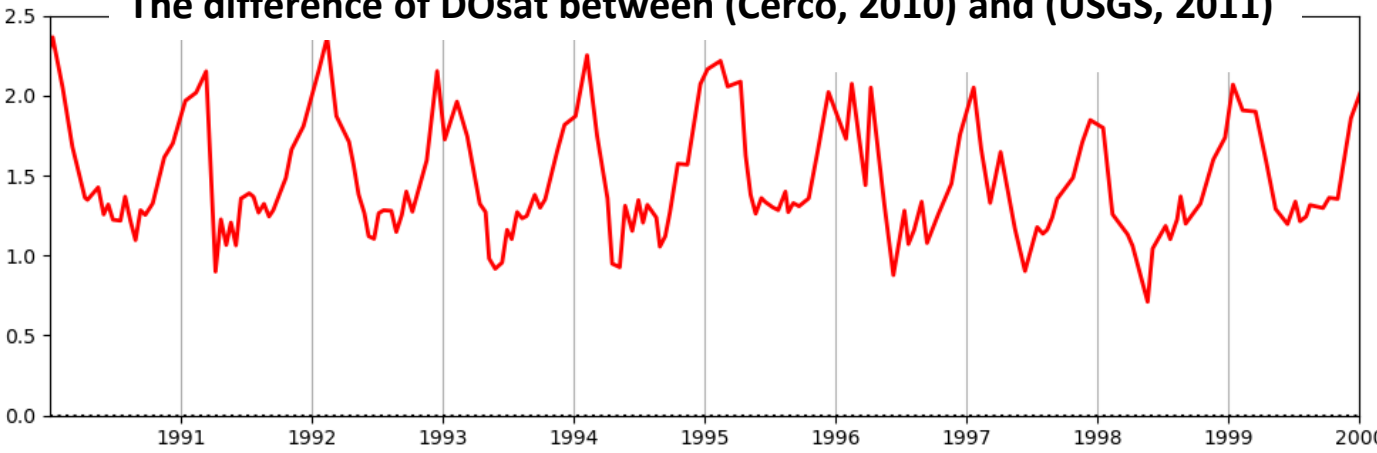
- Our previous modeling results overestimated DO especially in lower Bay.
- Thanks to the suggestion from Carl and Jiabi, we checked the formulation of saturation DO used in MBM, and found the formulation we used was outdated.

DO saturations (mg/L) from different formula @CB7.4





The difference of DOsat between (Cerco, 2010) and (USGS, 2011)



(Cerco, 2010): MBM was using

Saturation dissolved oxygen concentration diminishes as temperature and salinity increase. An empirical formula that describes these effects (Genet et al. 1974) is:

$$DOs = 14.5532 - 0.38217 \cdot T + 0.0054258 \cdot T^2 - CL \cdot (1.665 \times 10^{-4} - 5.866 \times 10^{-6} \cdot T + 9.796 \times 10^{-8} \cdot T^2) \quad (46)$$

(USGS, 2011) or (Cerco, 2019): CH3D is using

$$DO_o = \exp\left[-139.34411 + \frac{1.575701x10^5}{T} - \frac{6.642308x10^7}{T^2} + \frac{1.243800x10^{10}}{T^3} - \frac{8.621949x10^{11}}{T^4}\right], \quad (7)$$

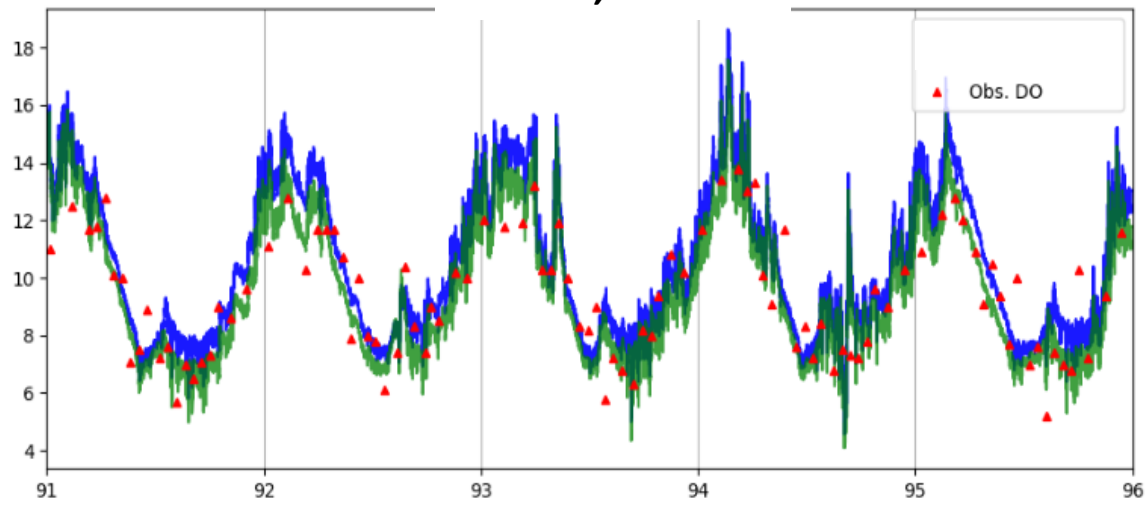
where T is the water temperature in Kelvin ($T = t(°C) + 273.15$). The Benson and Krause salinity factor is:

$$F_S = \exp\left[-S * \left(0.017674 - \frac{10.754}{T} + \frac{2140.7}{T^2}\right)\right], \quad (8)$$

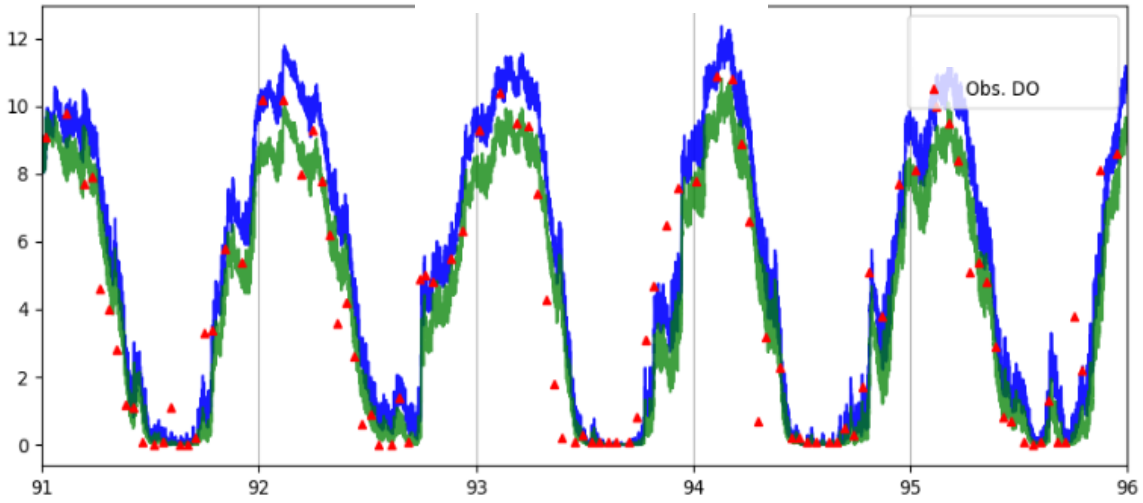
Improvement on DO simulation: time series

- When new formulation of DO saturation, DO simulation is greatly improved!

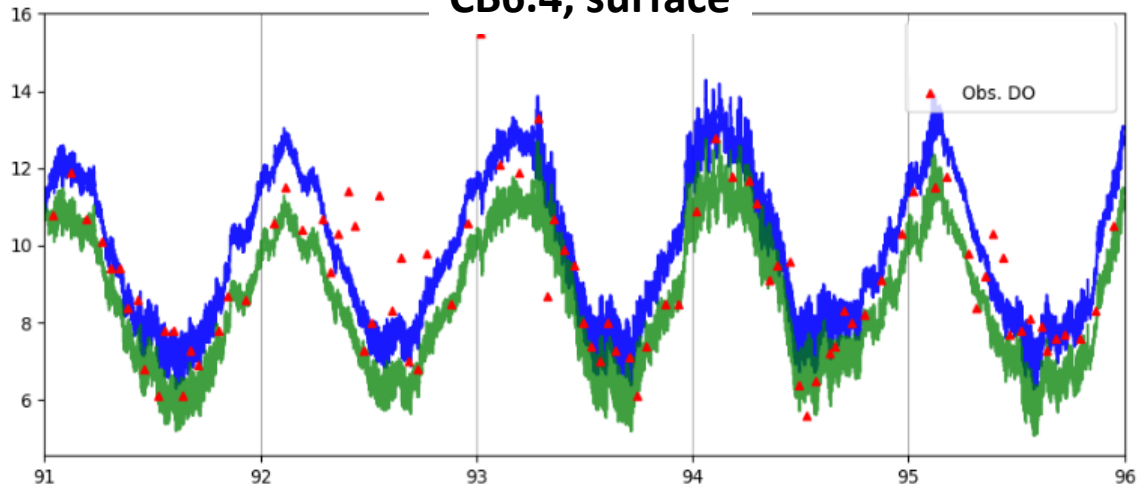
CB4.3C, surface



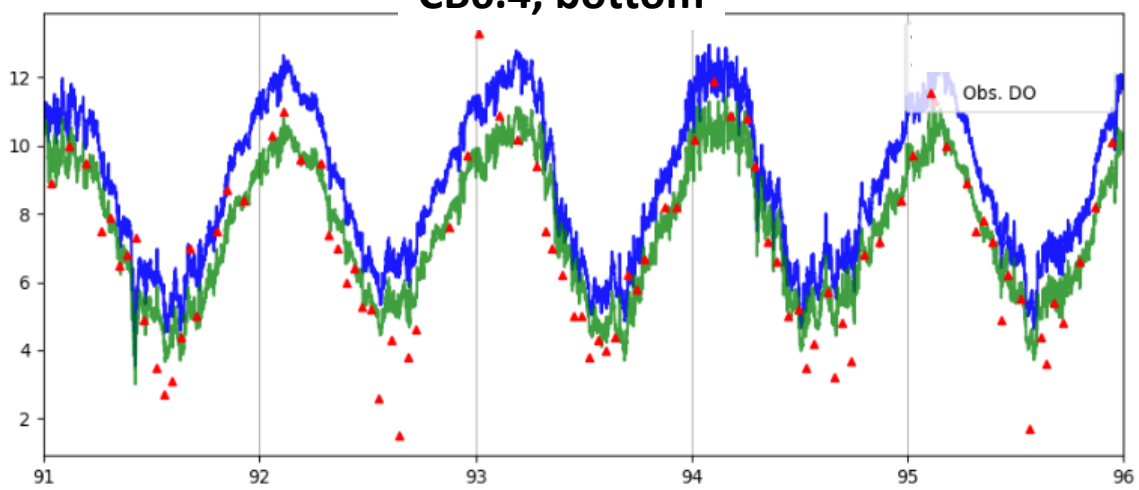
CB4.3C, bottom



CB6.4, surface

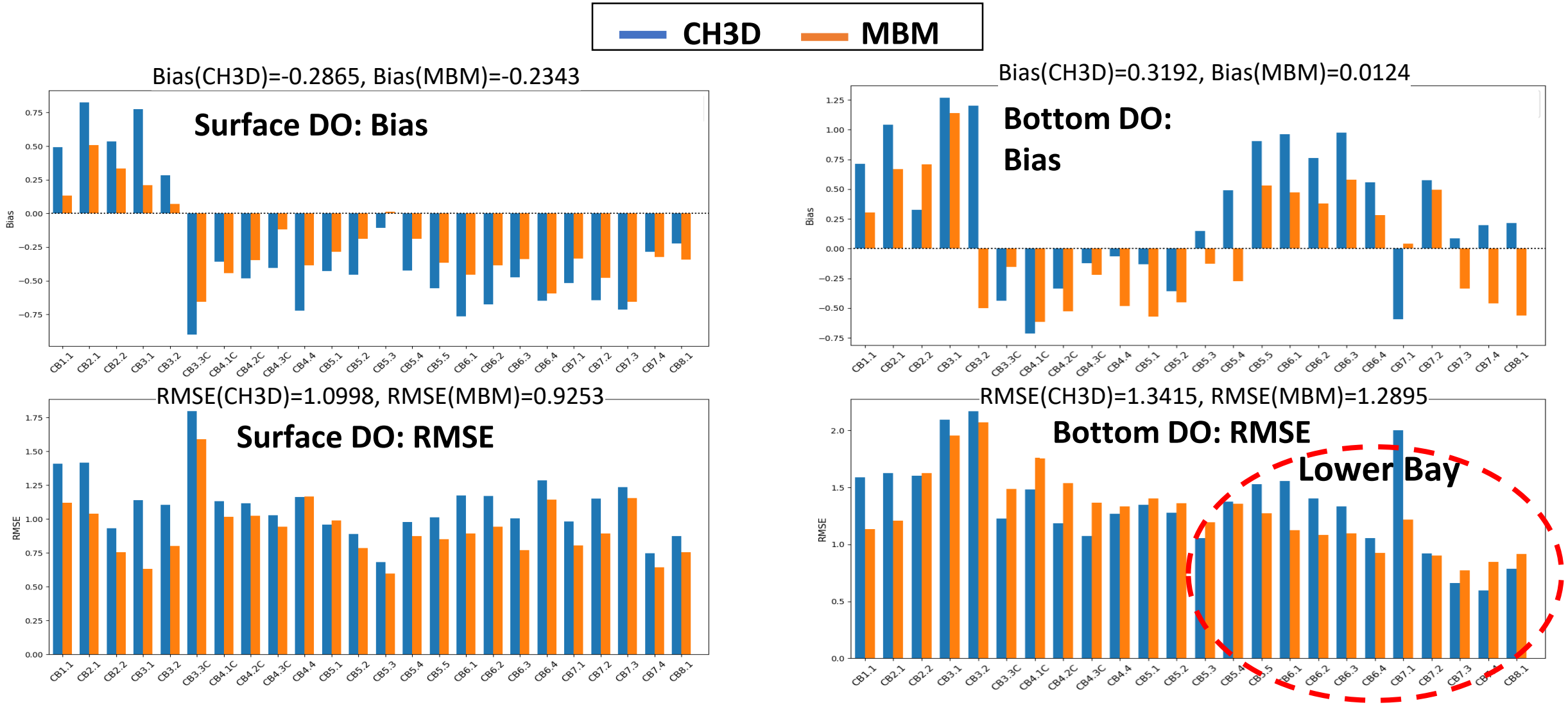


CB6.4, bottom



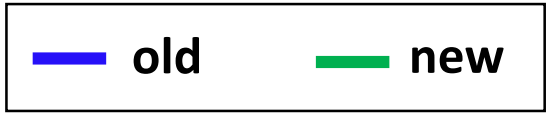
Improvement on DO simulation: model skill

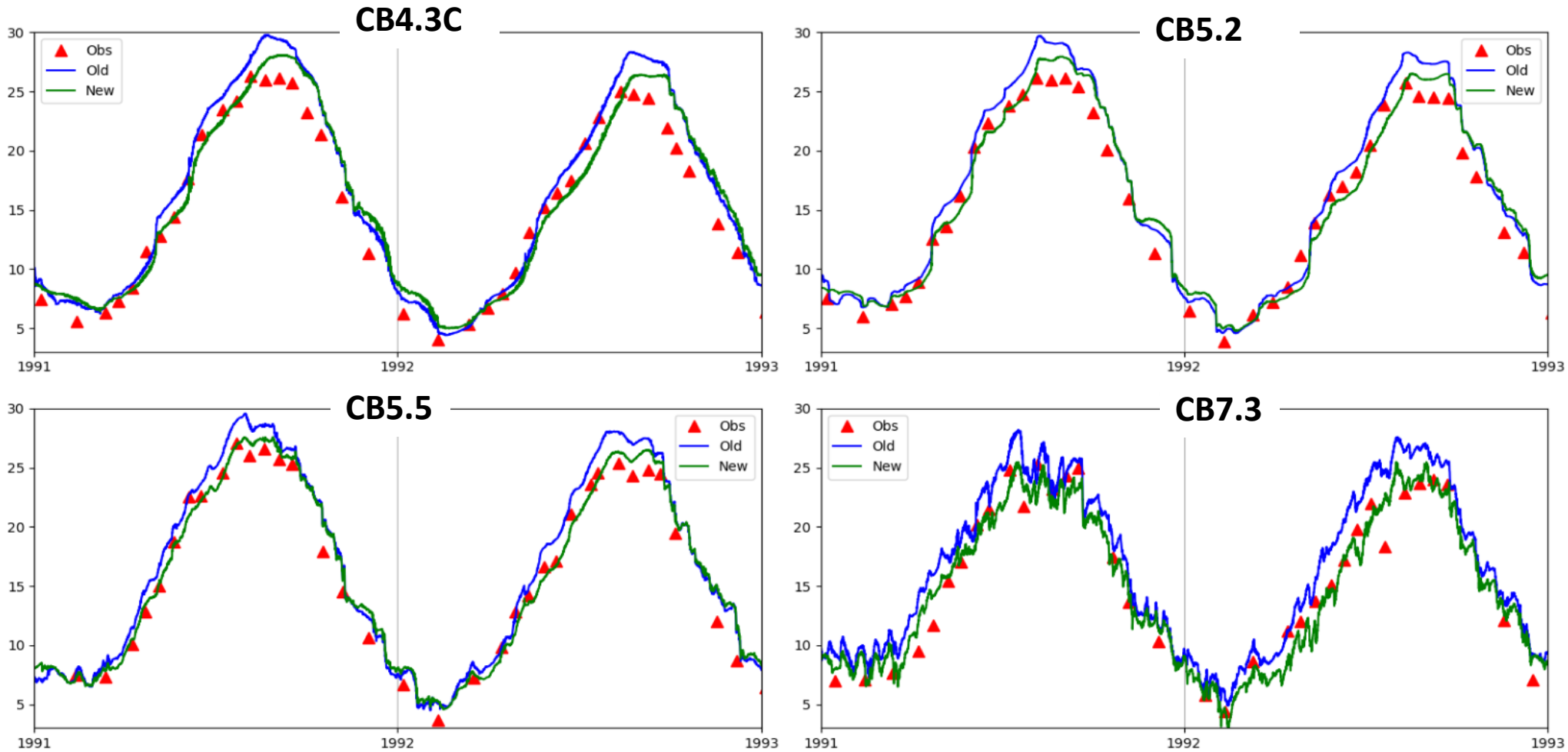
- By using new DO saturation formulation, the model skill of DO in MBM is comparable to CH3D
- Errors in two models share similar spatial pattern



Improvement #2: bottom temperature

- Bottom temperature was overestimated in MBM during warm months
- We improved the bottom temperature by adding the sediment-water heat exchange (buffering effects)





Adding Shoreline Erosion: Processing the Data

- Processed the shoreline erosion data, and added it into our database and incorporated it into our workflow.
- We prefer consolidated format for each category

Non-Point Source (NPS)

```
['time  : array(10957,), float64',
 'var   : list(19,)',
 'varname: dict(19,)',
 'units : dict(19,)',
 'sname : array(64,), <U13',
 'DO    : array(64, 10957), float32',
 'temp  : array(64, 10957), float32',
 'chla  : array(64, 10957), float32',
 'flow  : array(64, 10957), float32',
 'PO4   : array(64, 10957), float32',
 'NH4   : array(64, 10957), float32',
 'NO3   : array(64, 10957), float32',
 'TN    : array(64, 10957), float32',
 'TP    : array(64, 10957), float32',
 'ORGP  : array(64, 10957), float32',
 'ORGN  : array(64, 10957), float32',
 'PIP   : array(64, 10957), float32',
 'TC    : array(64, 10957), float32',
 'TSS   : array(64, 10957), float32',
 'sand  : array(64, 10957), float32',
 'silt  : array(64, 10957), float32',
 'clay  : array(64, 10957), float32',
 'phyto : array(64, 10957), float32']
```

Shoreline erosion (SHO)

```
['time  : array(10957,), float64',
 'var   : array(11,), <U5',
 'varname: dict(19,)',
 'units : dict(19,)',
 'sname : array(535,), <U20',
 'TN    : array(535, 10957), float32',
 'TP    : array(535, 10957), float32',
 'ORGP  : array(535, 10957), float32',
 'ORGN  : array(535, 10957), float32',
 'PIP   : array(535, 10957), float32',
 'TC    : array(535, 10957), float32',
 'TSS   : array(535, 10957), float32',
 'sand  : array(535, 10957), float32',
 'silt  : array(535, 10957), float32',
 'clay  : array(535, 10957), float32']
```

❖ **To streamline the workflow, we would prefer a simpler format for all watershed loadings as shown in *NPS* and *SHO*.**

- ✔ **One file (*.shp, *.mat, or *.npz, or netCDF): much reduced file size**
- ✔ **Large data matrices instead of many small files**
- ✔ **Easier for data processing (e.g. search)**

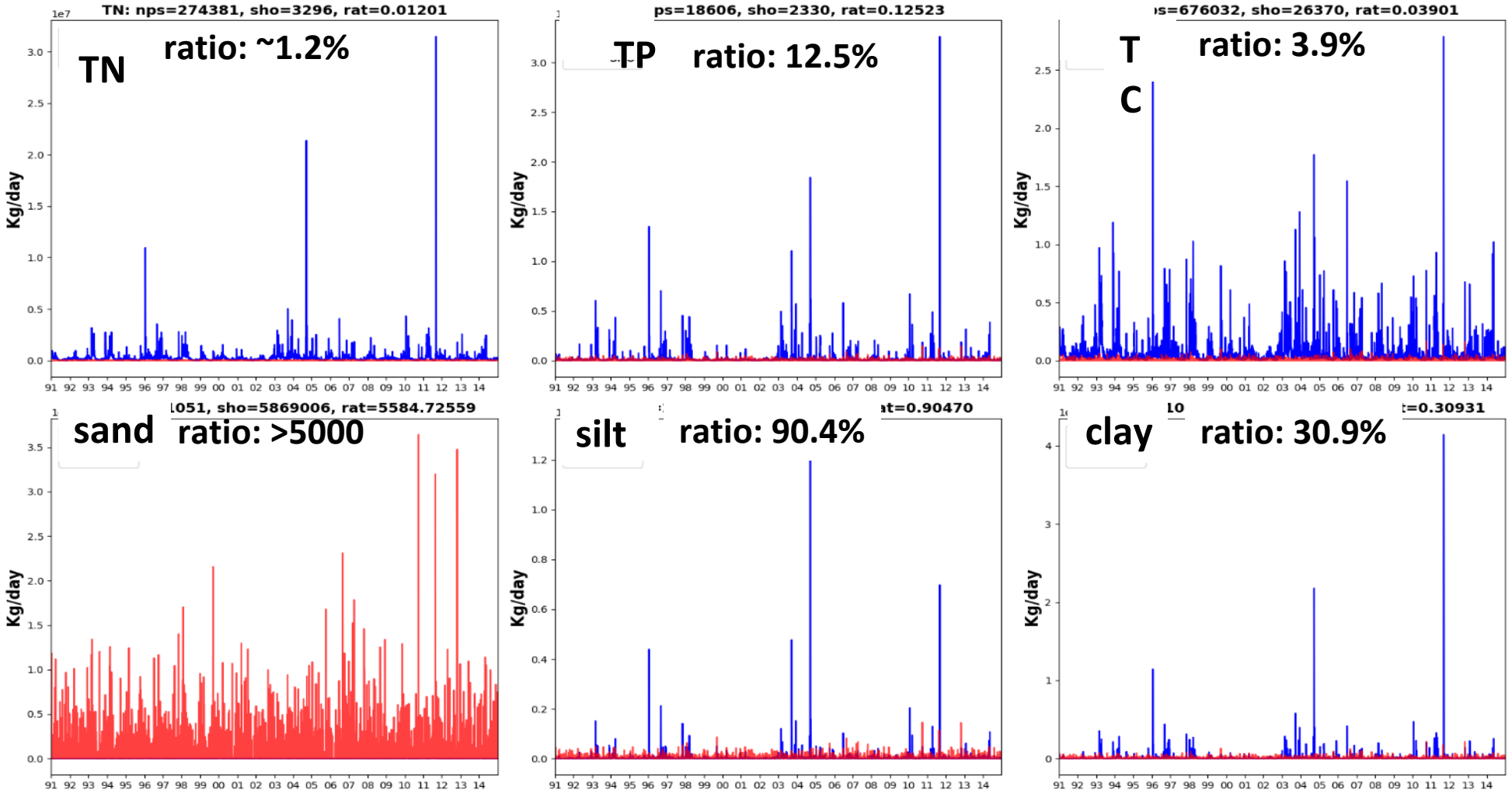
Shoreline erosion vs non-point source

- The ratios of (TN, TP, TC) between SHO and NPS are about 1.2%, 12.5% and 3.9%, respectively.
- The SHO sediment (sand, silt, clay) loading is important

NPS

SHO

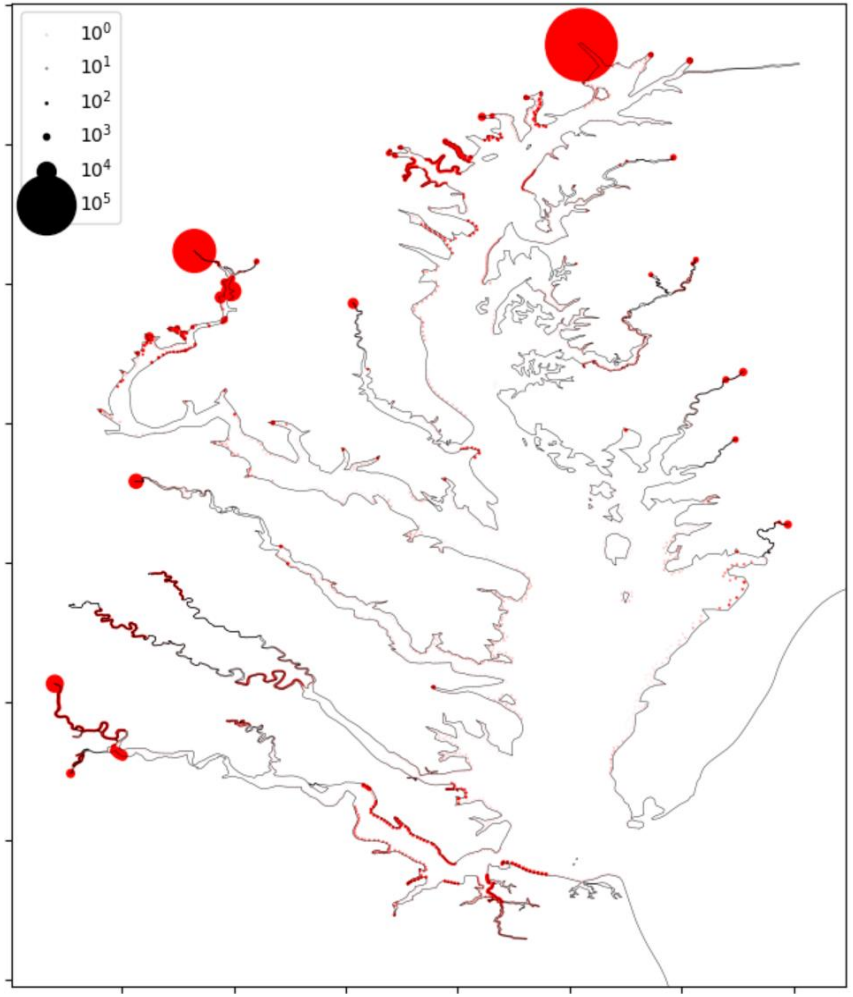
Time Series of loading from NPS and SHO.



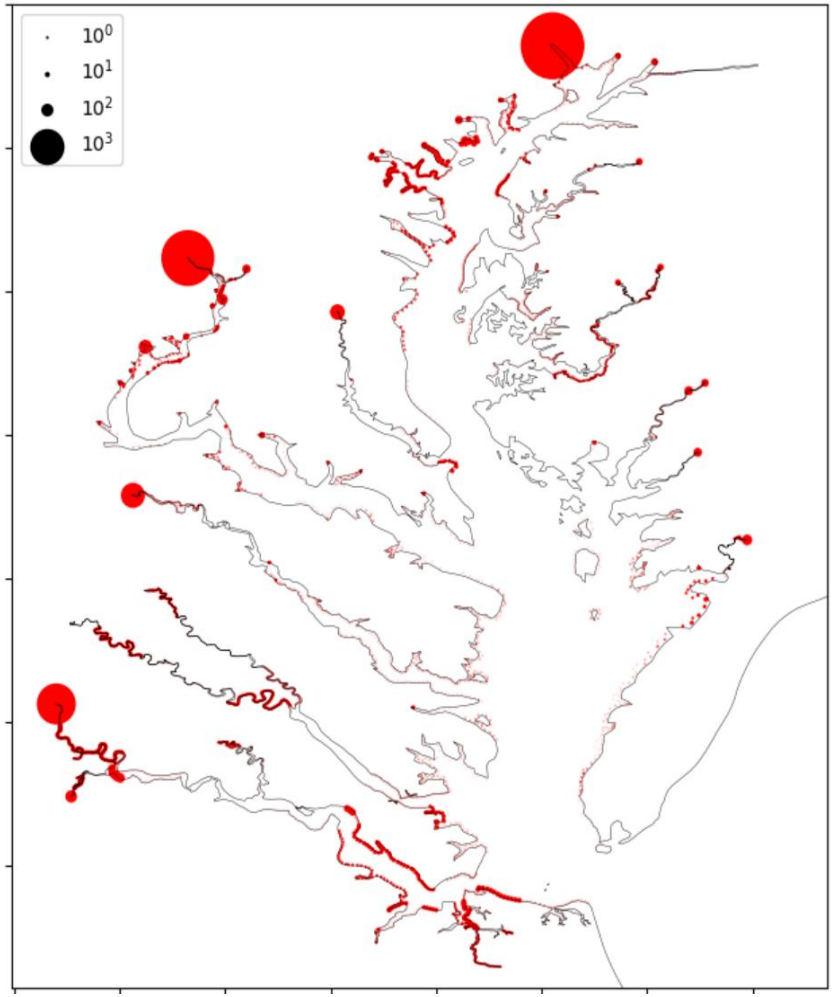
Spatial distribution of nutrient loading in MBM

- We combine the loadings from NPS, PS and SHO, and then distribute them along MBM grid boundary
- New algorithm for distribution takes advantage of high resolution and works robustly across the domain

TN loading (Kg/day)



TP loading (Kg/day)



Processing the atmospheric loading

- Processed the atmospheric loading data (from CMAQ), added it into our database and incorporated it into our workflow.
- At the moment, atmospheric loading is interpolated from CMAQ onto CH3D and then interpolated onto MBM grid
- We prefer original netcdf outputs from CMAQ (which also reduces file size).

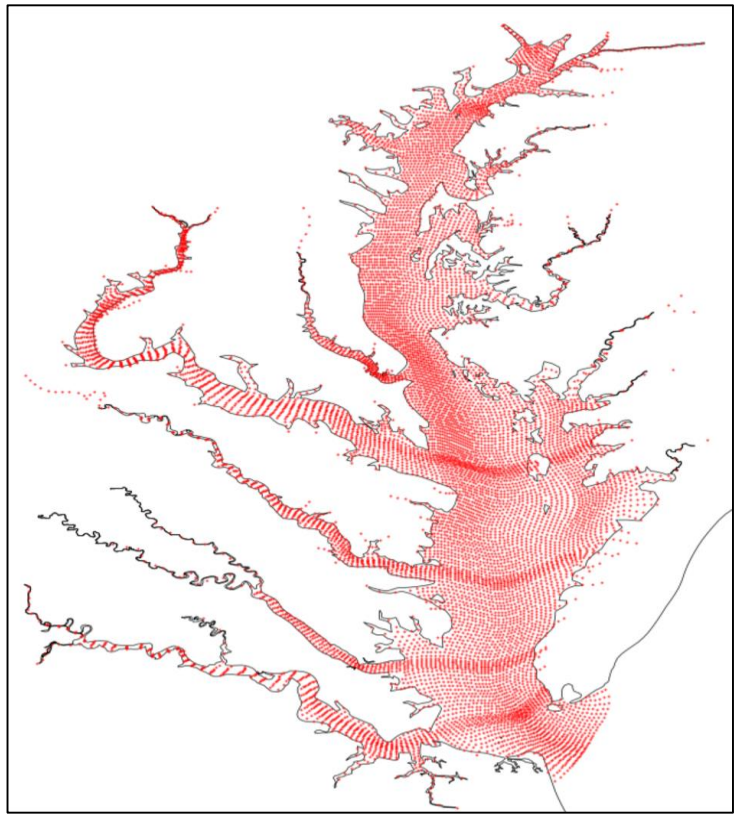
Non-Point Source (NPS)

```
['time  : array(10957,), float64',
 'var   : list(19,)',
 'varname: dict(19,)',
 'units : dict(19,)',
 'sname : array(64,), <U13',
 'DO    : array(64, 10957), float32',
 'temp  : array(64, 10957), float32',
 'chla  : array(64, 10957), float32',
 'flow  : array(64, 10957), float32',
 'PO4   : array(64, 10957), float32',
 'NH4   : array(64, 10957), float32',
 'NO3   : array(64, 10957), float32',
 'TN    : array(64, 10957), float32',
 'TP    : array(64, 10957), float32',
 'ORGP  : array(64, 10957), float32',
 'ORGN  : array(64, 10957), float32',
 'PIP   : array(64, 10957), float32',
 'TC    : array(64, 10957), float32',
 'TSS   : array(64, 10957), float32',
 'sand  : array(64, 10957), float32',
 'silt  : array(64, 10957), float32',
 'clay  : array(64, 10957), float32',
 'phyto : array(64, 10957), float32']
```

Atmospheric Loading (ATM)

```
['NH4   : array(11064, 10957), float32',
 'NO3   : array(11064, 10957), float32',
 'ORGN  : array(11064, 10957), float32',
 'ORGP  : array(11064, 10957), float32',
 'PO4   : array(11064, 10957), float32',
 'area  : array(11064,), float64',
 'cell  : array(11064,), float32',
 'time  : array(10957,), float64',
 'unit  : dict(6,)',
 'var   : list(6,)',
 'varname: dict(6,)',
 'x     : array(11064,), float64',
 'y     : array(11064,), float64']
```

Atmospheric Loading on CH3D grid



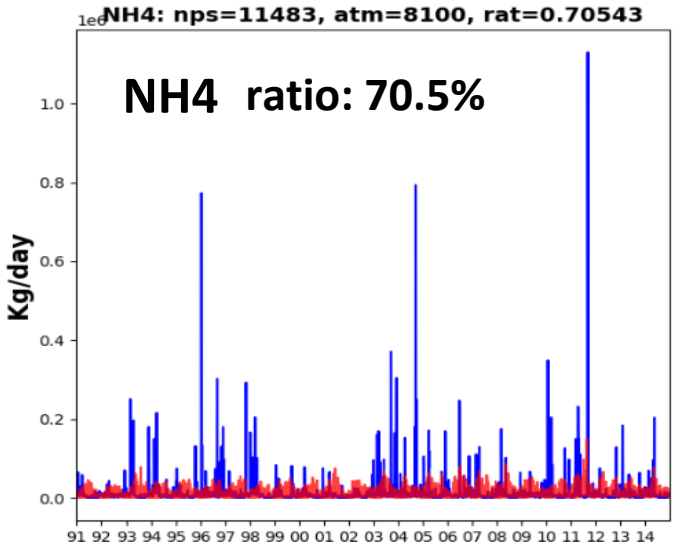
ATM vs NPS loadings

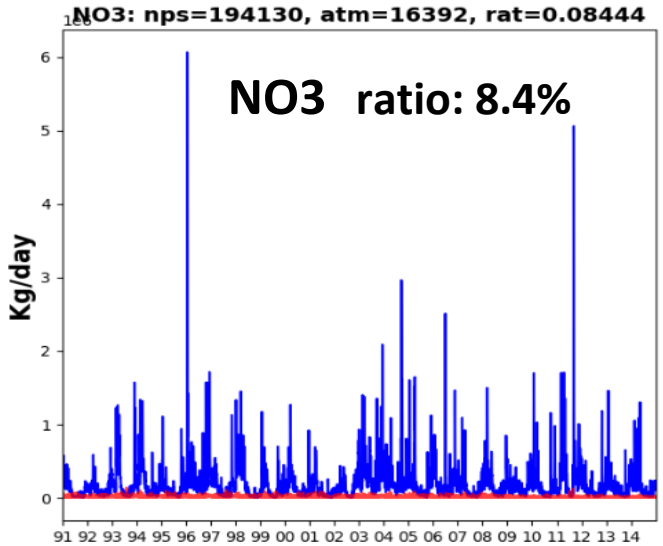
- The ATM loading is relatively significant when compared with NPS loading for both N and P.
- For TN and TP, the ratios between ATM and NPS are about 9.6% and 12.3%.
- Note that ATM loading is applied on a large surface area compared to NPS

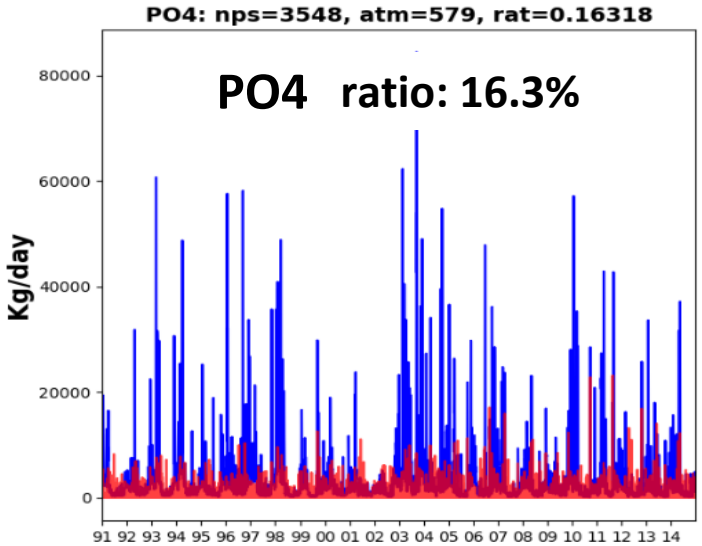
NPS

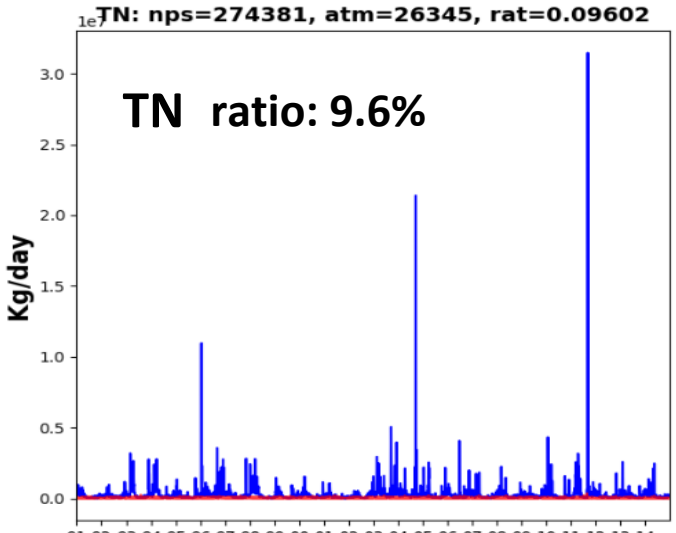
ATM

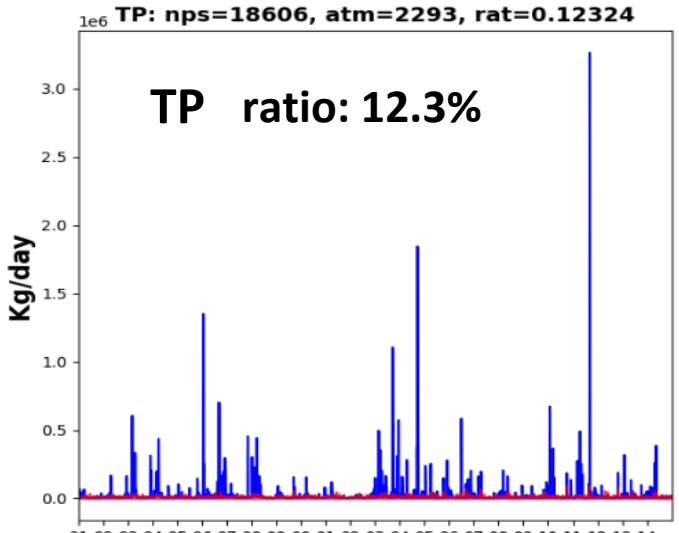
Time Series of loading from NPS and SHO.







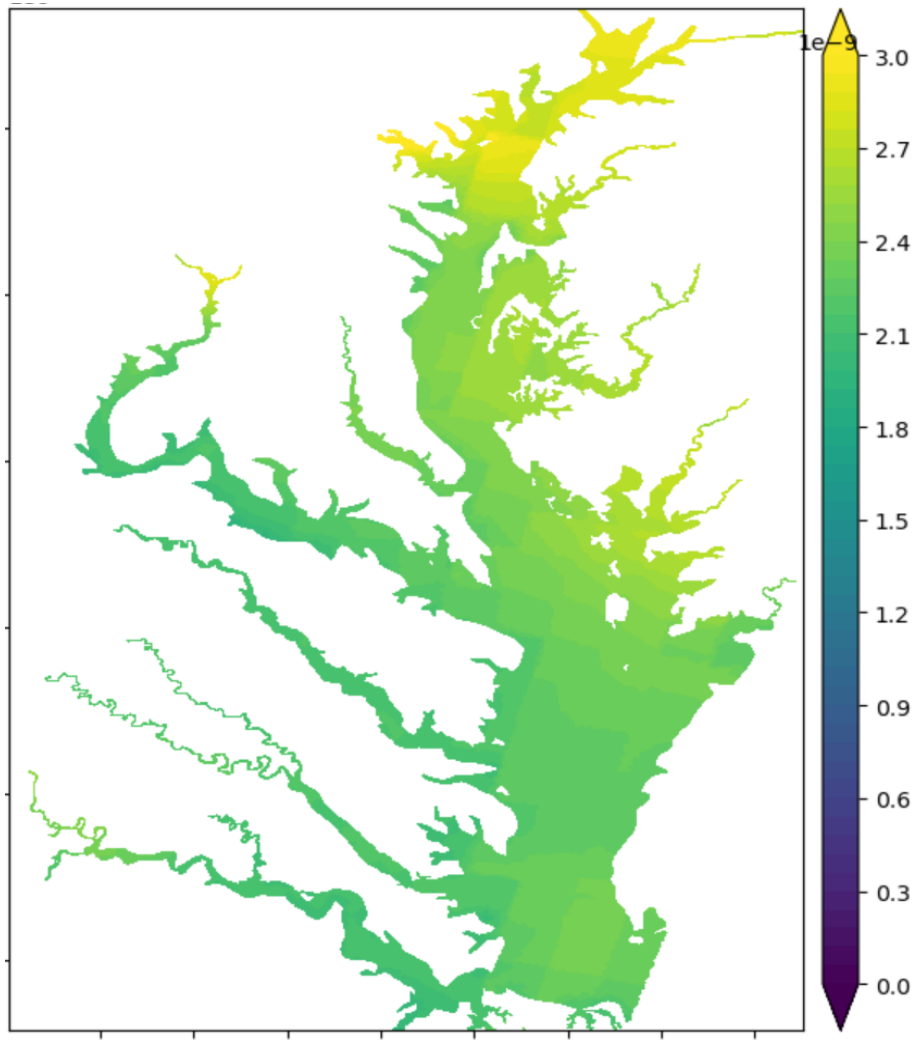




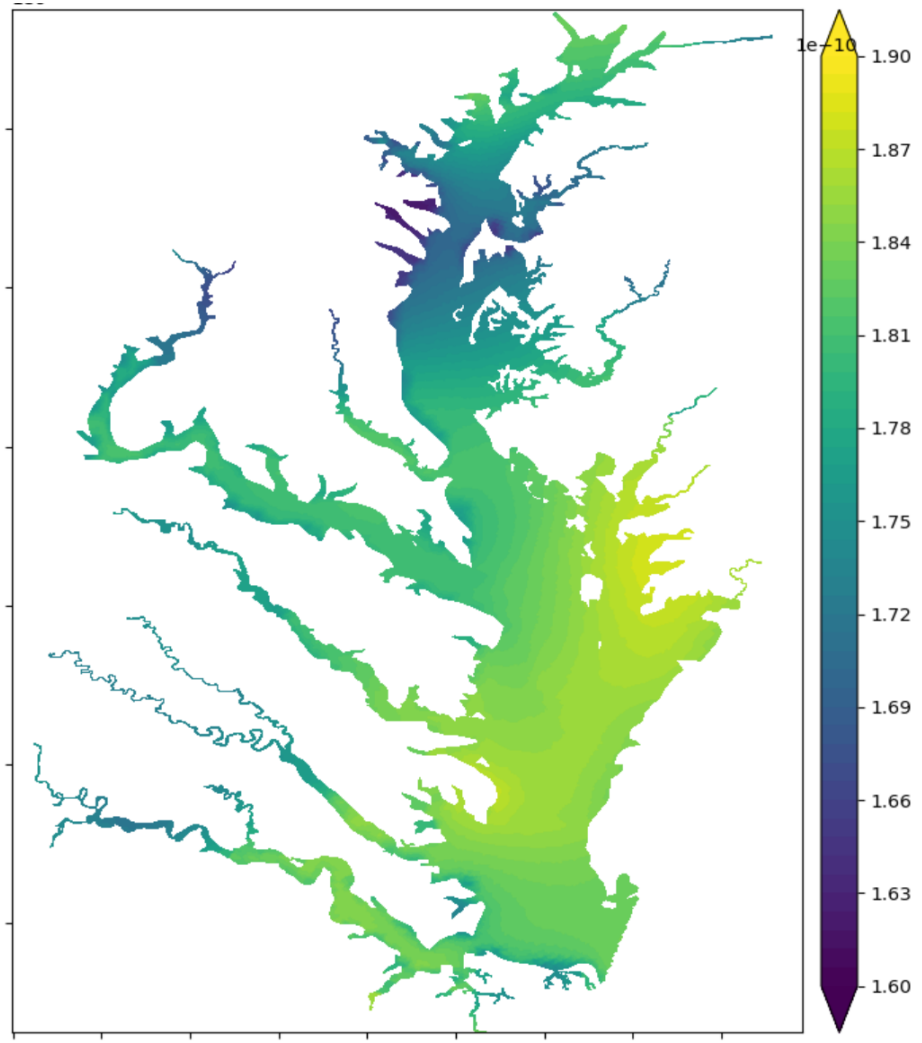
Spatial distribution of atmospheric deposition in MBM

- High TN upper Bay, and high TP in mid-lower bay

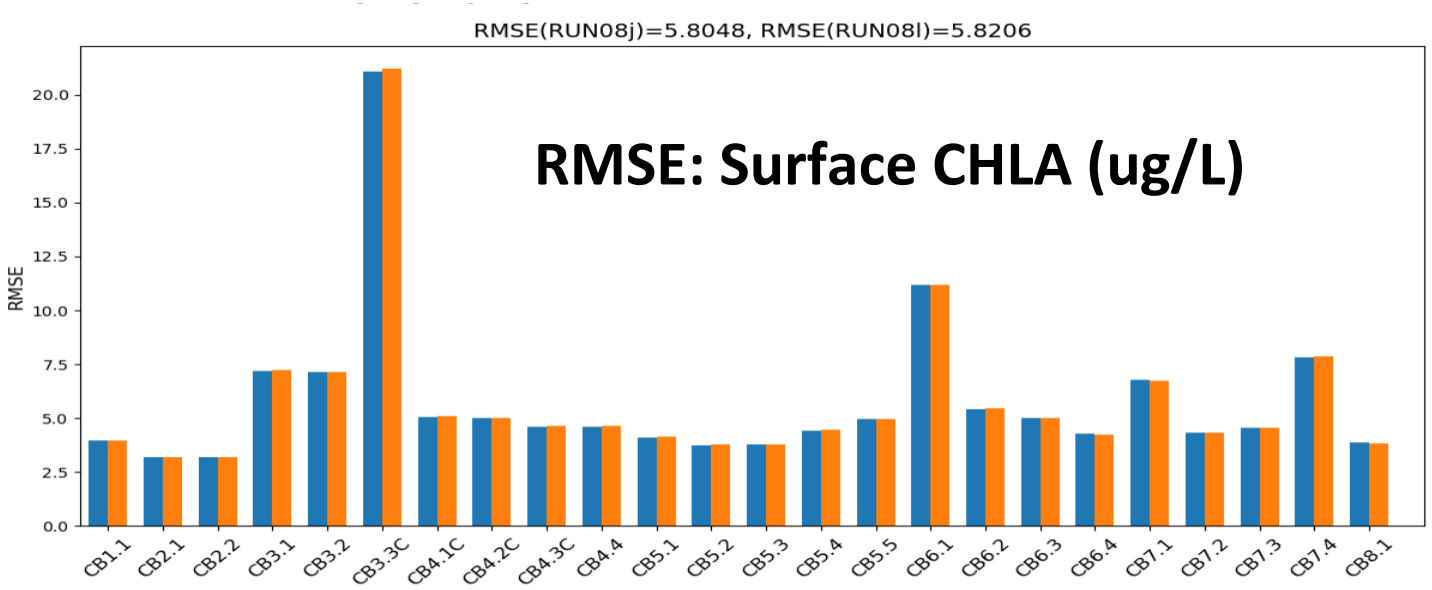
TN loading ($g.m^{-2}.day^{-1}$)

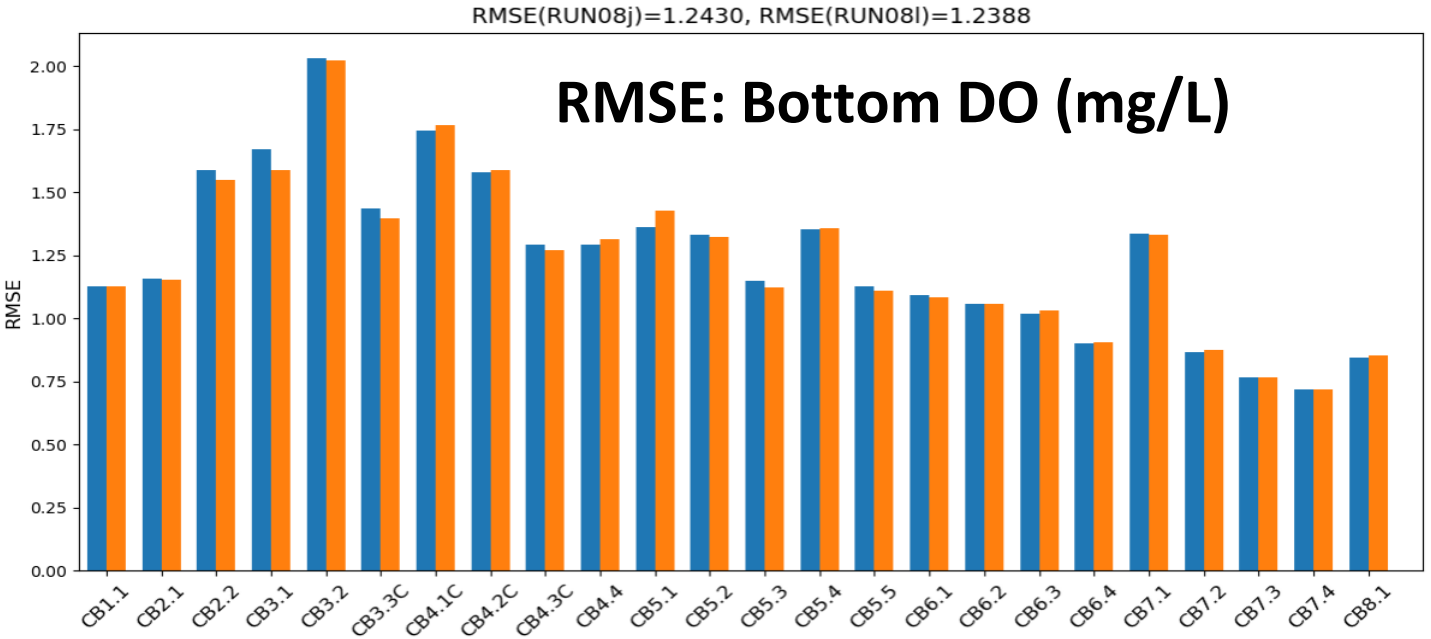


TP loading ($g.m^{-2}.day^{-1}$)



With both shoreline erosion and atmospheric loadings added





Latest model results (including 4 slow reactive nutrients) are largely consistent with previous results

Updating workflow: Database

- The entire modeling workflow is streamlined in python from grid generation/model setup all the way to results analysis/visualization.
- Recent updates include:
  - Added additional 4 ICM variables: **SRPOC**, **SRPON**, **SRPOP**, **PIP**
  - Updated databases of shoreline and atmospheric loadings
  - Updated python algorithm/script to generate ICM model input files

MBM Databases

```
bdir='/sciclone/data10/wangzg/CBP/setup_files/'       #dir of setup files
p.saldata   = bdir+'ChesBay_Salinity_climatology.nc' #salinity data
p.cbpdata   = bdir+'CBP_WQData.npz'                   #CBP water quality database
p.source    = bdir+'load_p6_v3.npz'                   #CBP watershed sources
p.atmdep    = bdir+'atm_load.npz'                     #CBP atmospheric deposition
p.hycom     = bdir+'HYCOM/Data'                       #HYCOM database
p.elev_noaa = bdir+'noaa_elev_msl.npz'                #noaa elevation database
p.elev_adjt = bdir+'elev_adjust_v0.npz'               #database on bnd to adjust elevation
p.sflux     = bdir+'sflux_narr_subdomain'             #sflux database
p.NEFSC     = bdir+'NEFSC_climatology.npz'            #nutrient database in ocean
p.bedfrac   = bdir+'ROMS_bedfrac.npz'                 #ROMS sediment bedfraction
p.WW3       = bdir+'WW3'                              #WW3 wave forcing
p.hydro_out = bdir+'hydro/RUN07b/outputs'             #hydro_out for offline ICM model
p.outdir    = '/sciclone/pscr/{}/CBP'.format(os.environ['USER']) #parental direcotry of outputs
p.station   = bdir+'bp/station_CB.bp'                 #station information for ICM model
```

Updating workflow: Model Inputs

simulation period

choose grid

choose sub-models

hydrodynamic inputs

ICM model inputs

sediment model inputs

wave model inputs

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

p.base='../RUN08k'
p.grid_dir='/sciclone/data10/wangzg/CBP/grid/v2' #direct

p.flag['ICM']= 10  #ICM model (1: 21 variables; 10: 21-v
p.flag['SED']= 0  #SED3D model
p.flag['WWM']= 0  #Wave model

p.flag['elev2D.th.nc']          = 0 #hydro
p.flag['TEM_3D.th.nc']          = 0 #hydro
p.flag['TEM_nu.nc']             = 0 #hydro
p.flag['SAL_3D.th.nc']          = 0 #hydro
p.flag['SAL_nu.nc']             = 0 #hydro
p.flag['sflux']                 = 0 #hydro
p.flag['albedo.gr3']            = 0 #hydro
p.flag['drag.gr3']              = 0 #hydro
p.flag['watertype.gr3']         = 0 #hydro
p.flag['diffmin.gr3']           = 0 #hydro
p.flag['diffmax.gr3']           = 0 #hydro
p.flag['shapiro.gr3']           = 0 #hydro
p.flag['TEM_nudge.gr3']         = 0 #hydro
p.flag['SAL_nudge.gr3']         = 0 #hydro
p.flag['tvd.prop']              = 0 #hydro
p.flag['windrot_geo2proj.gr3'] = 0 #hydro
p.flag['rough.gr3']             = 0 #hydro,SED
p.flag['bctides.in']            = 0 #hydro,SED,ICM
p.flag['hotstart.nc']           = 0 #hydro,SED,ICM
p.flag['source.nc']             = 0 #hydro,SED,ICM
p.flag['ICM_3D.th.nc']          = 0 #ICM
p.flag['istation.in']           = 0 #ICM
p.flag['ICM_nudge.gr3']         = 0 #ICM
p.flag['ICM_nu.nc']             = 0 #ICM
p.flag['ICM_sflux.th.nc']       = 1 #ICM
p.flag['SED_hvar_1.ic']         = 0 #SED
p.flag['SED_hvar_2.ic']         = 0 #SED
p.flag['SED_hvar_3.ic']         = 0 #SED
p.flag['SED_hvar_4.ic']         = 0 #SED
p.flag['bed_frac_1.ic']         = 0 #SED
p.flag['bed_frac_2.ic']         = 0 #SED
p.flag['bed_frac_3.ic']         = 0 #SED
p.flag['bed_frac_4.ic']         = 0 #SED
p.flag['bedthick.ic']           = 0 #SED
p.flag['SED_nudge.gr3']         = 0 #SED
p.flag['hgrid_WWM.gr3']         = 0 #WWM
p.flag['wwmbnd.gr3']            = 0 #WWM
p.flag['WW3-forcing']           = 0 #WWM
```

- In the recent update, we added or updated the algorithm/script for the following model input files

  - **hotstart.nc**
    initial condition for 4 new variables

  - **source.nc**
    algorithm for adding shoreline erosion

  - **ICM_3D.th.nc**
    boundary condition of new variables

  - **ICM_nudge.gr3, ICM_nu.nc**
    relaxation algorithm for ICM variables in coastal ocean

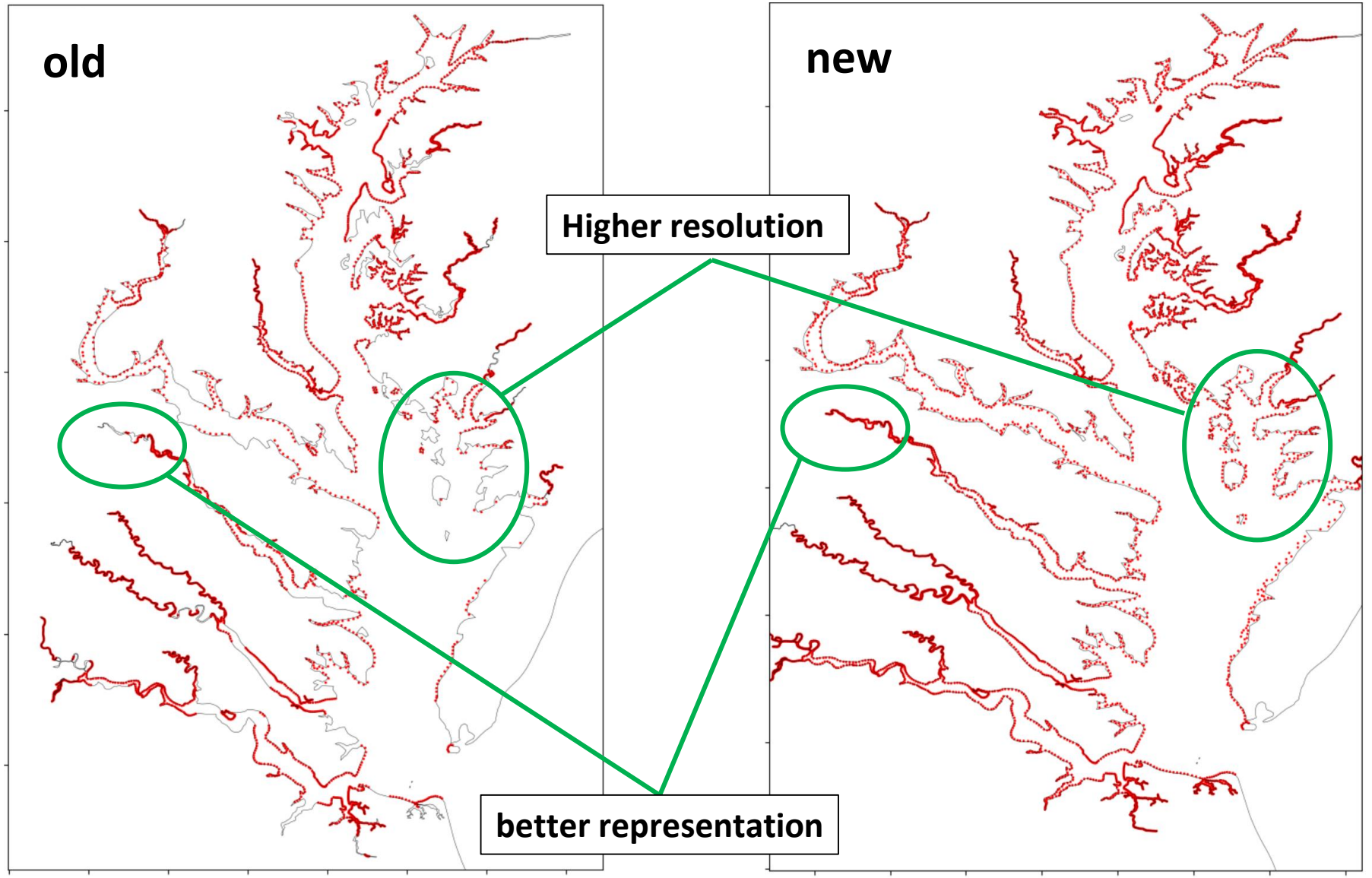
  - **ICM_sflux.th.nc**
    algorithm to add atmospheric loading

  - **ICM_param.nc**
    For spatially varying ICM parameters

Updating algorithm for watershed loading allocation

- Revamped algorithm to take advantage of the high resolution used in MBM grid
  - Fixed an issue in our algorithm in dealing with watersheds of multi-segments.

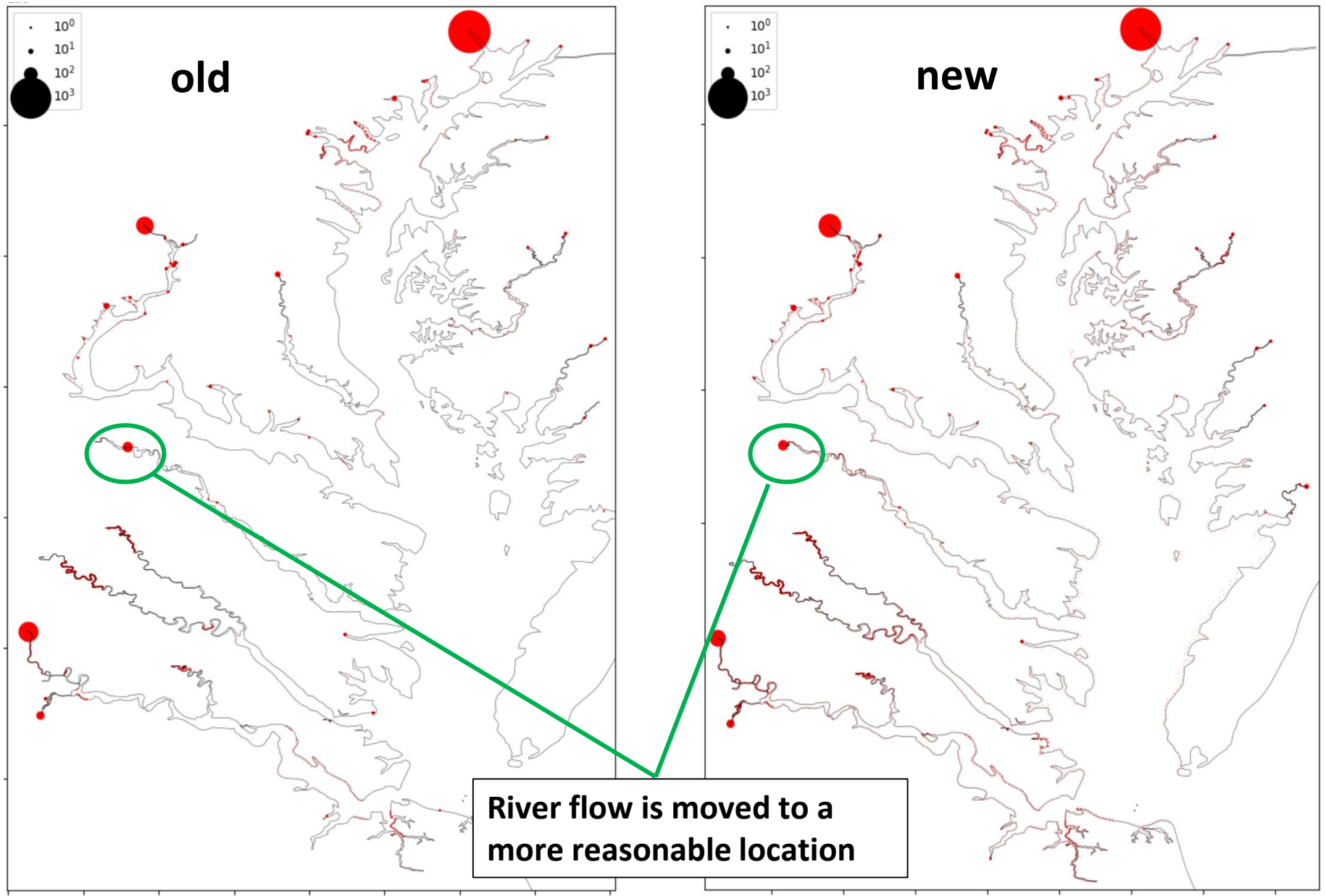
Source locations in MBM



Updating watershed loading allocation

❖ The inflow locations from watershed are moved to more reasonable places

Flow rates from watershed loading (m^3/s)

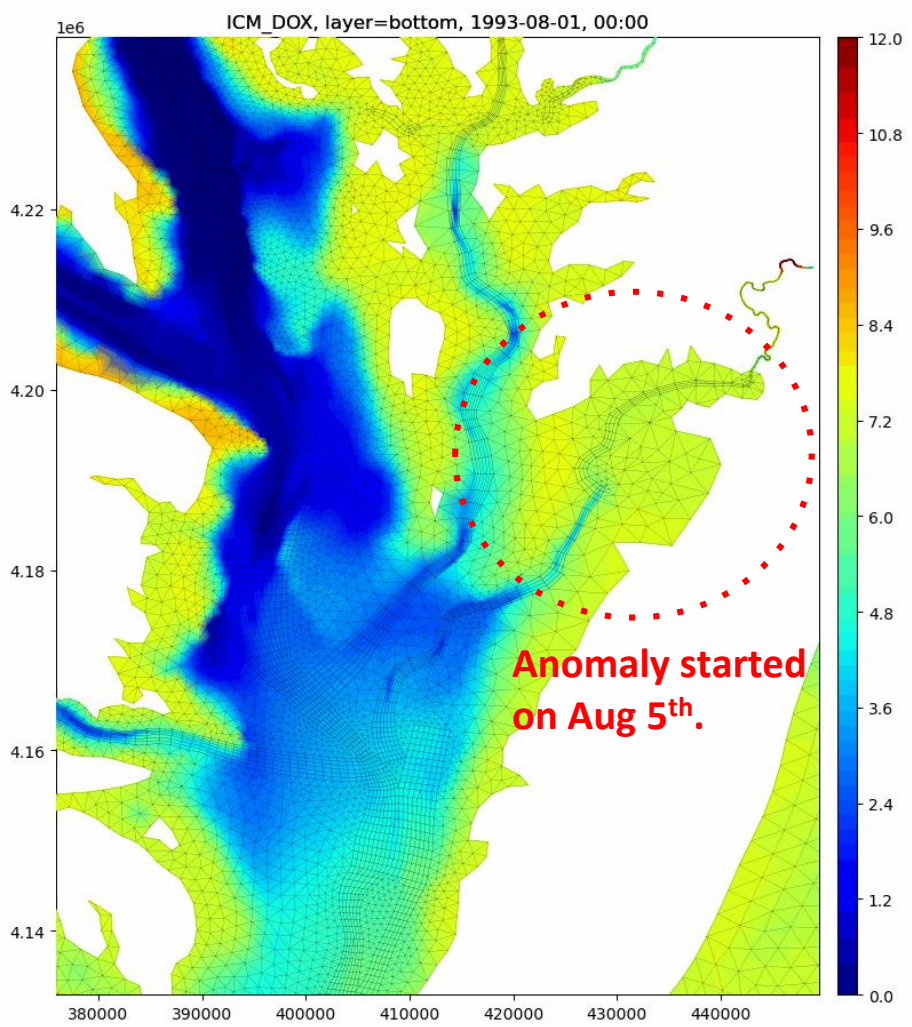


Remaining issue: high concentrations in watershed loadings

- ❖ Concentration = mass divided by flow
- ❖ If the flow rate is close to 0, concentration can be very high, which is not realistic
- ❖ In general, total loading combined with very small flow is problematic

Example: DO problem

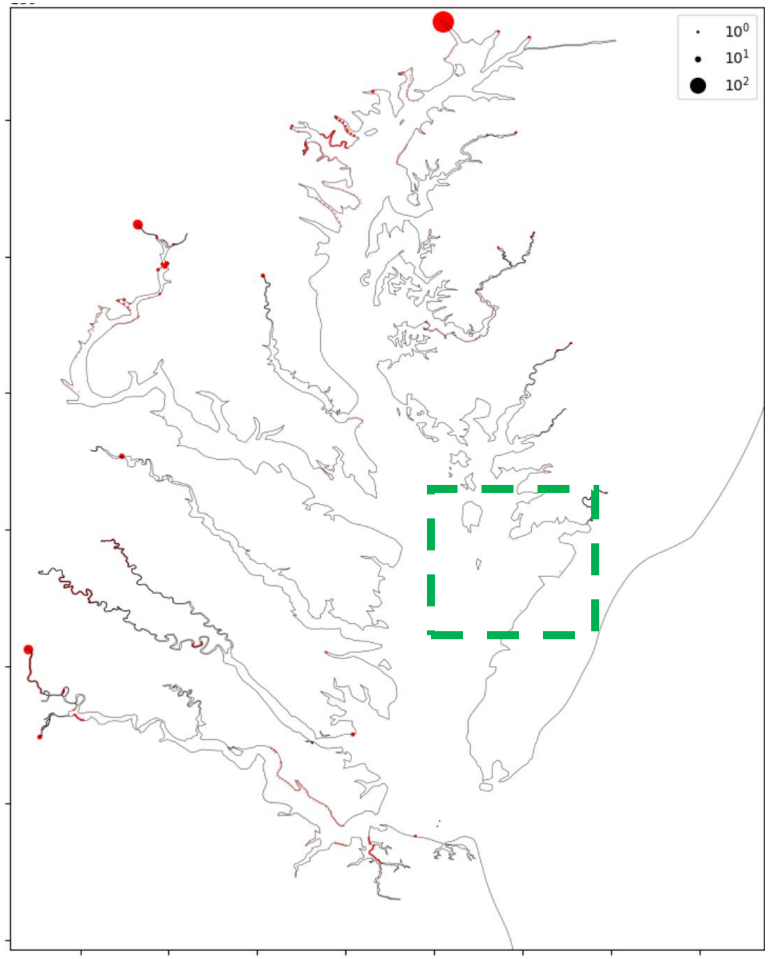
In MBM simulation, we identified some DO anomalies; e.g. in Pocomoke Sound during Aug-Sep period, 1993.



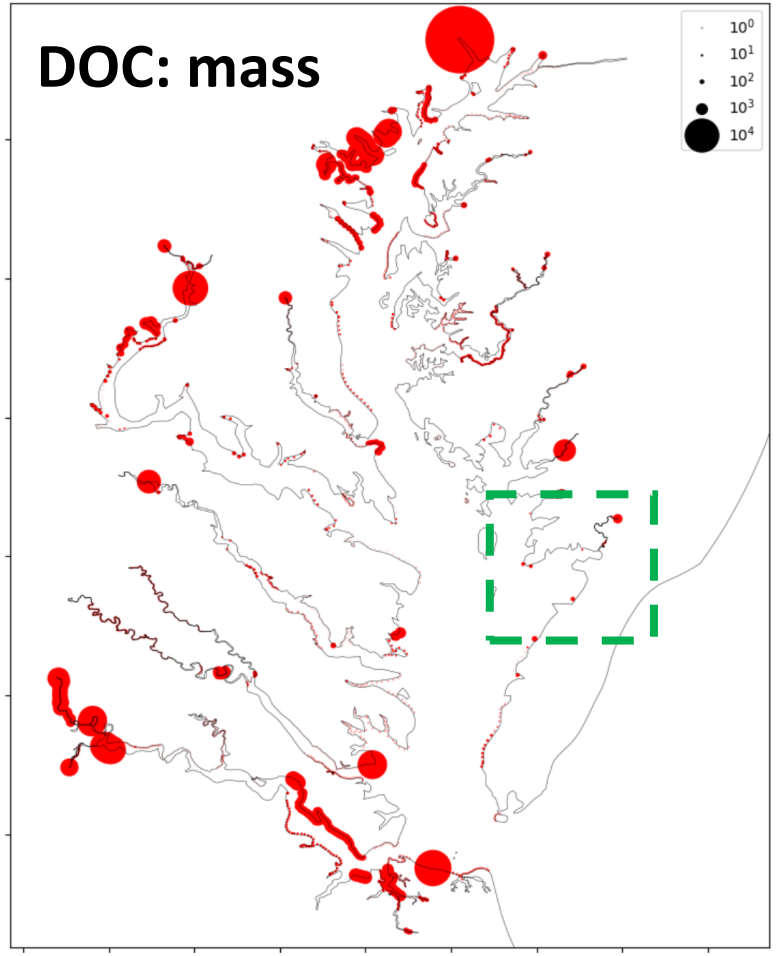
Remaining issue: high concentrations in watershed loadings

❖ In addition to checking flow and mass loading, it's important to also check concentration

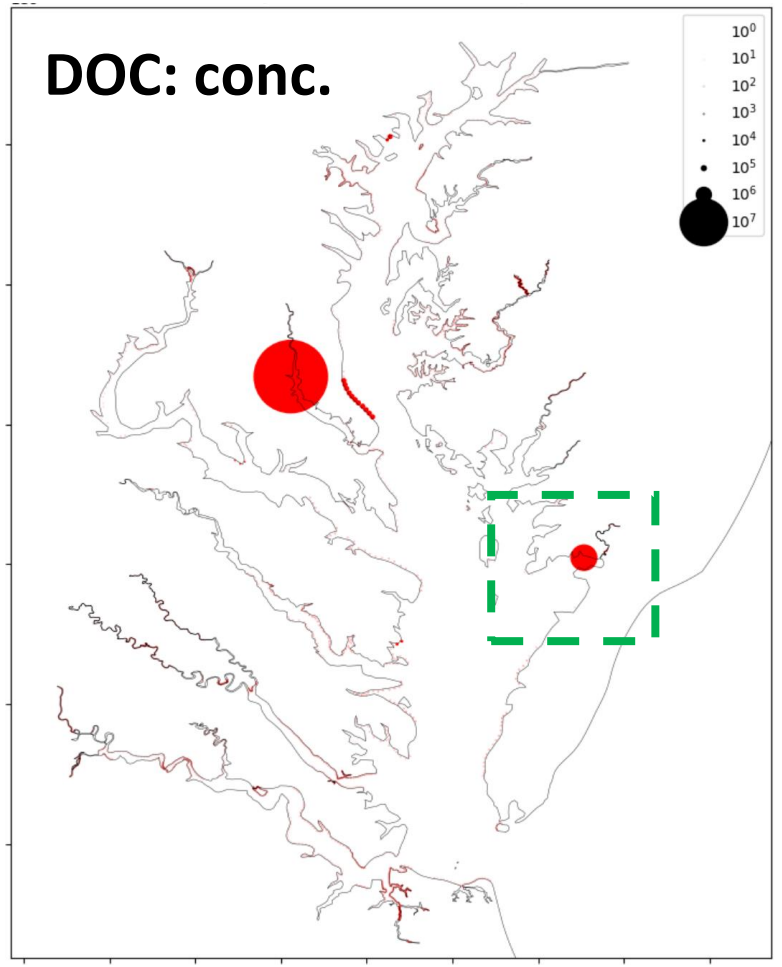
Flow (m^3/s)



Loading (Kg/day)



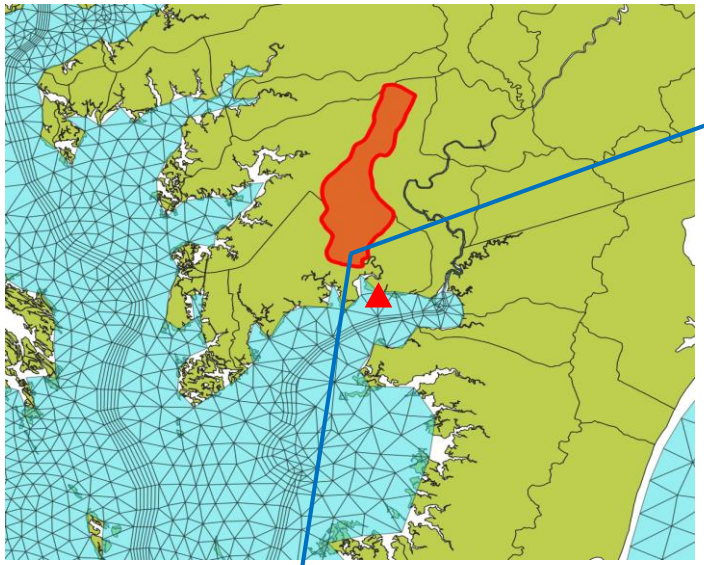
Concentration (mg/L) is unreasonable



Remaining issue: high concentrations in watershed loadings

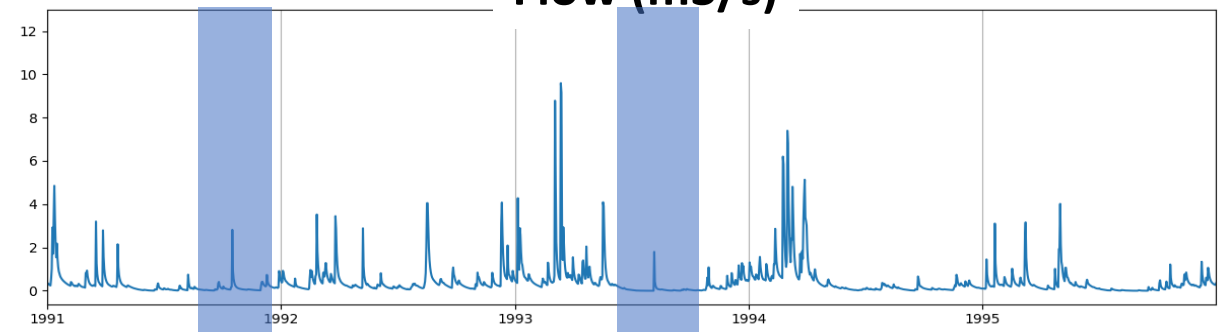
- Unrealistically large concentrations are problematic
  - Most ICM formulas (e.g. DO) are based on concentrations $dDO = -o2n \cdot Nit \cdot NH4 - o2c \cdot K_{HR} \cdot DOC - K_{COD} \cdot COD$ …
  - High concentration from loading would result in unreasonable numbers from those formulas
  - High resolution exacerbates the issue

watershed: Pocomoke

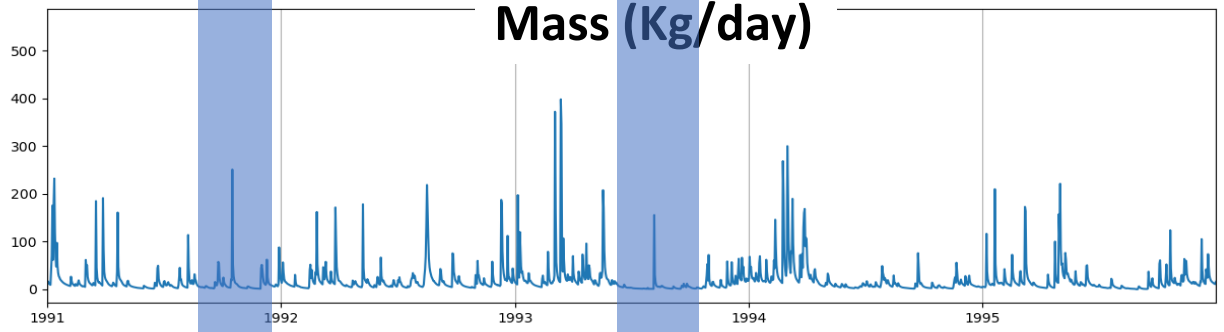


Feature	Value
MajBas	Eastern Shore of Chesapeake Bay
MajMin	EL
RiverSeg	EL1_6000_0001
Minor	L
MinBas	Lower Eastern Shore
RiverName	Marumsco Creek
ST	MD
TidalWater	N
PRECIP	N
FIPS_NHL	N24039
LndRvrSeg	N24039EL1_6000_0001
CBSEG_92	POCMH_MD

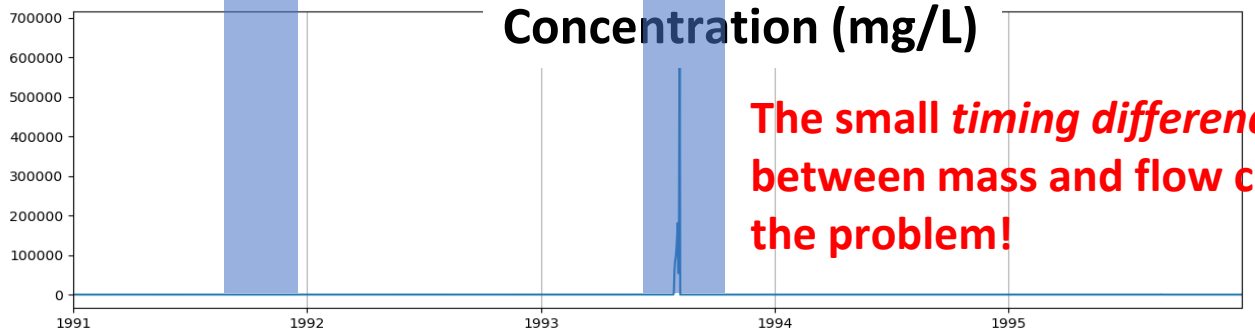
Flow (m3/s)



Mass (Kg/day)



Concentration (mg/L)



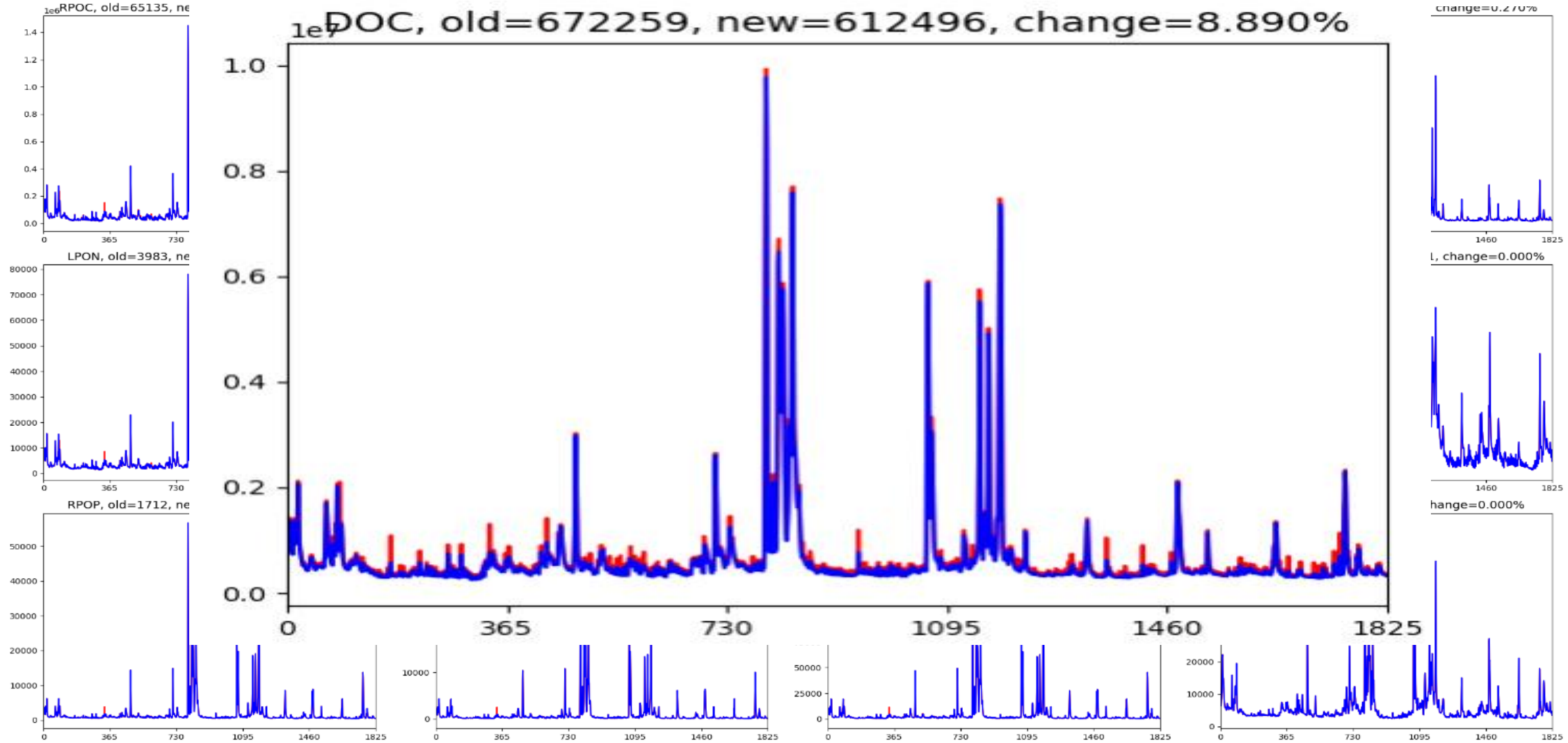
The small *timing difference* between mass and flow caused the problem!

A work-around: total loading with and without concentration correction

old new

- A work-around by imposing a max concentration and impose a baseline flow
  - Alternatively, we can ‘dilute’ the concentration using a large enough area…
- But it has implications for TMDL (by reducing mass loading)
- So it’d be good to correct this in the watershed model

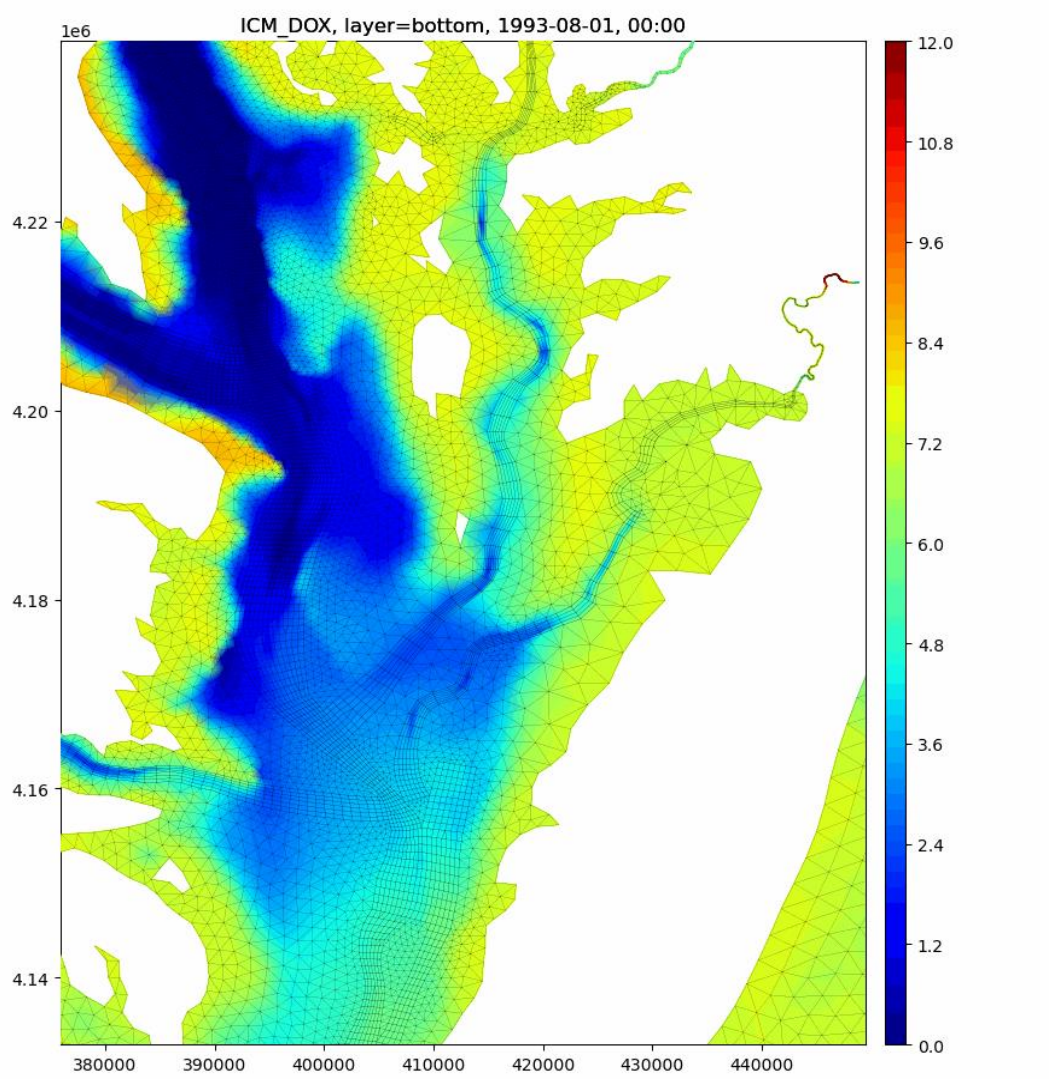
Total Loading



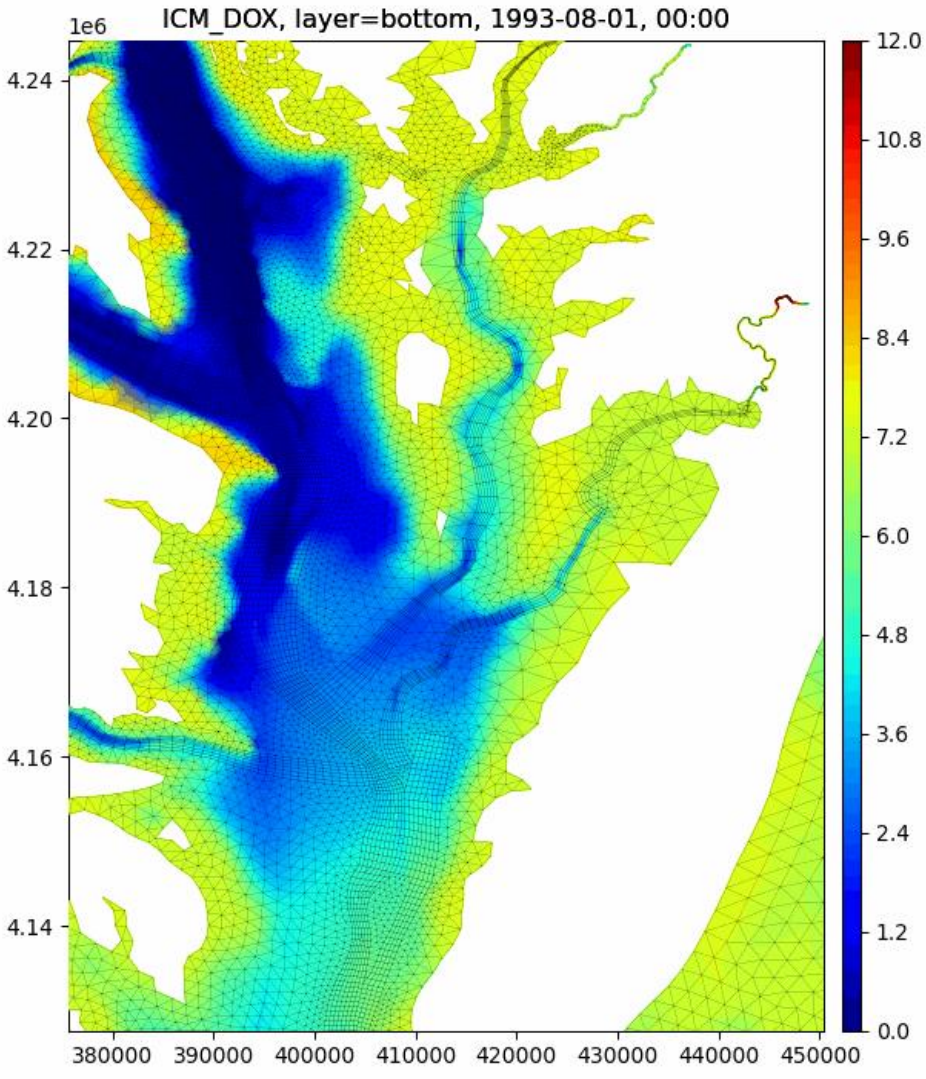
DO simulation with work-around

- After fixing the DOC concentration, DO results seem reasonable
- But this correction has implication for TMDL

Before Correction



After Correction



Summary & future plan

- ✓ We have significantly improved the simulation of DO and bottom temperature
- ✓ Added missing shoreline erosion and atmospheric deposition loadings
- ✓ Updated modeling workflow

- ✓ Identified issues in watershed loadings
  - ➢ Working with watershed modeling team to find a solution
- ✓ Will further improve the model skills for nutrients (e.g. PO4)
- ✓ Will work on missing sub-modules (living resources)