Recent Development in ROMS Data Assimilation and Applications: 4D-Var Revisited

Hernan G. Arango[1], Andrew M. Moore[2], and John L. Wilkin[1]

[1]Rutgers University, DMCS, New Brunswick, NJ, USA
[2]University of California, Santa Cruz, CA, USA



CP-TT/DA-TT Workshop, ECMWF, Reading, UK
October 28, 2026

4D-Var Revisited

- Split 4D-Var across multiple executables, with mixed resolution and mixed precision.
  - Grid decimation
- IODA-type multiple observation files, enhanced NetCDF-4 with first-level grouping.
  - Refactoring of 4D-Var observation processing
- Redesign of $H(x)$ and $H^T(x)$ operators
  - Control vector variable changes
  - Area-averaging and/or time-averaging operators
- Multi-scale background-error covariance modeling
  - Implicit horizontal and vertical diffusion operators
  - Spatially-varying horizontal correlation length scales
- Variational bias correction.
- Correlated observation error (non-diagonal R).
- Nested 4D-Var



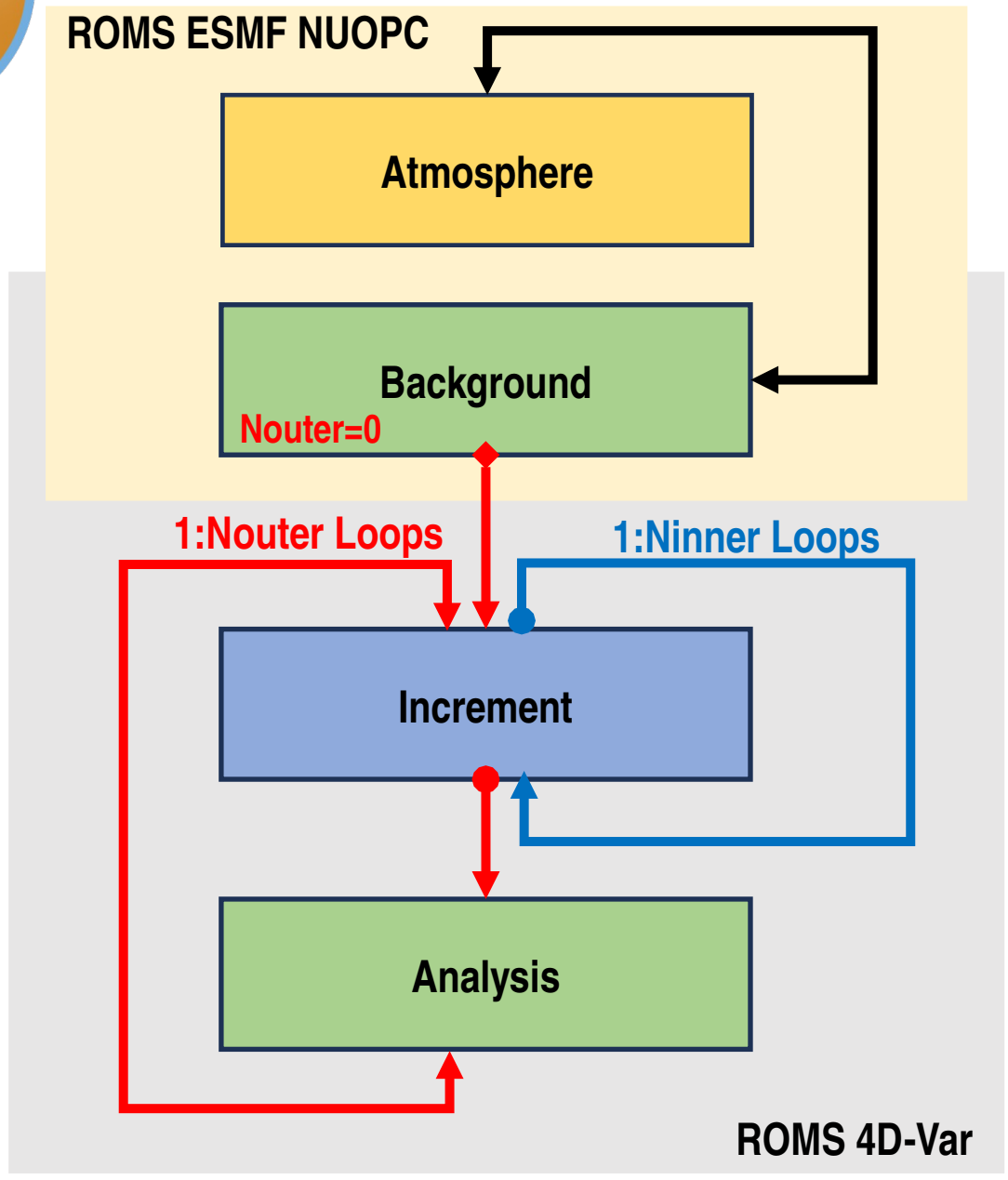
ROMS Data Assimilation Evolution



ROMS Adjoint Group
A. Bennett's OSU 2002 IOM Summer School
Corvallis, OR

Year	History
2000	ONR funded the ROMS Adjoint Group to develop adjoint-based, advanced data assimilation and sensitivity algorithms (GST propagators, 4D-Var, etc.).
2007	February 6: First release of adjoint-based algorithms with TLM, RPM, and ADM kernels (ROMS 2.3).
2011	Moore *et al.*, 2011a,b,c: ROMS 4D-Var algorithms, review papers, *Prog. Oceanogr.*, doi:10.1016/j.pocean.2011.05.004, doi:10.1016/j.pocean.2011.05.003, and doi:10.1016/j.pocean.2011.05.005.
2014	Gürol *et al.*, 2014: RBL4D-Var algorithm paper, *Q.J.R. Meteorol. Soc.*, doi:10.1002/qj.2150.
2020	Moore *et al.*, 2020: DART's EAKF and 4D-Var paper, Prog. Oceanogr., doi:10.1016/j.pocean.2020.102450.
2021	March 10: JEDI's AMS Short Course (virtual) and June 21-26: Arango and Moore attended the 6th JEDI Academy (virtual).
2023	Moore *et al.*, 2023: ROMS Saddle-Point Algorithm (SP4D-Var), *Ocean Model.*, doi:10.1016/j.ocemod.2023.102262.
2023	Arango *et al.*, 2023: ROMS nested 4D-Var algorithm, *Ocean Model.*, doi:10.1016/j.ocemod.2023.102201.
2025	June 30: Public release of the ROMS-JEDI interface at github.com/myroms/roms-jedi.
2026	Major update to native ROMS 4D-Var framework (ROMS 4.3).

Split 4D-Var Framework via Bash Script



Background Phase: Executable A (roms_nl)

- Operates at a higher horizontal grid resolution.
- Computes fine and coarse NLM kernel trajectories.
- Uses grid decimation to write trajectories from the fine to the coarse grid needed by the TLM and ADM kernels.
- Evaluates H(x) at a higher spatial resolution.
- Can be integrated into an ATM-OCN coupling system.
- May include nested computational grids.
- Operates in either single or double precision.

Increment Phase: Executable B (roms_da)

- Runs at coarser horizontal grid resolution.
- Executes minimization steps within inner loops of the data assimilation cycle (RPCG Lanczos).
- Iterates the ADM and TLM kernels.
- Computes a coarser increment during each outer loop.
- Usually operates in double precision (recommended).

Analysis Phase: Executable A (roms_nl)

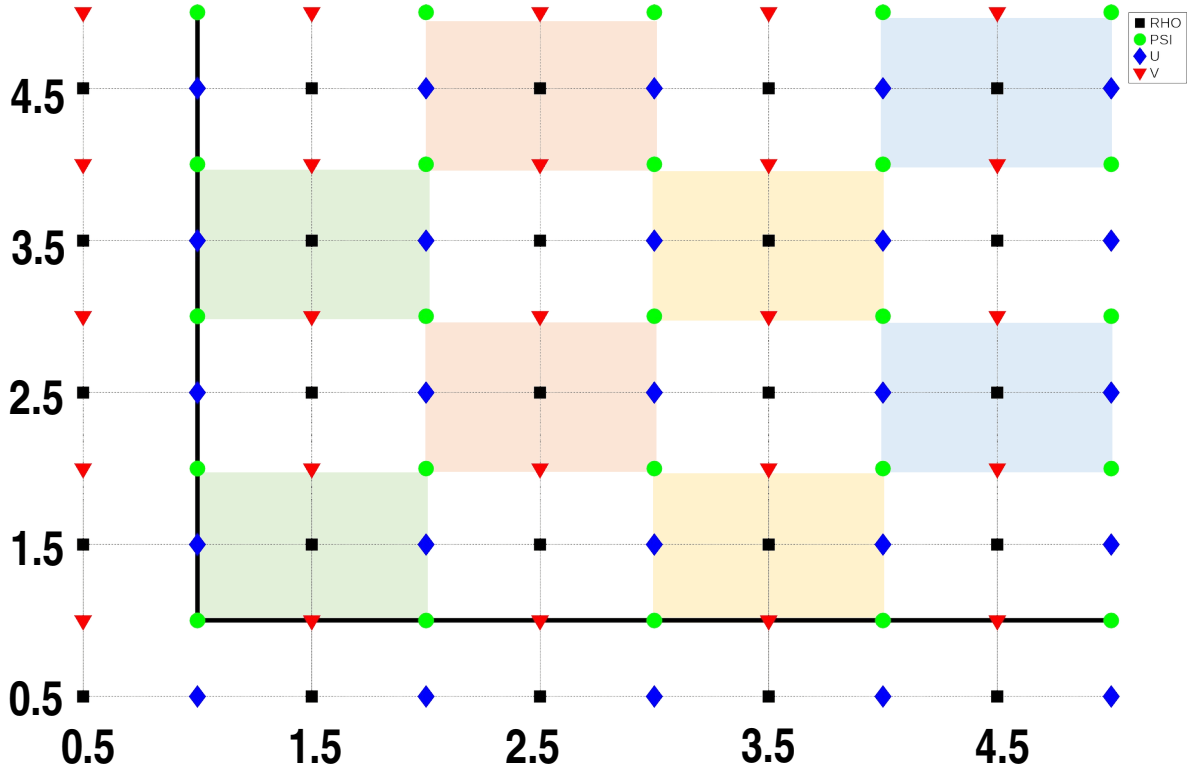
- Applies coarse-to-fine resolution mapping to the analysis state vector.
- Executes the NLM kernel.

Grid Decimation: Reduction by a Factor of Two

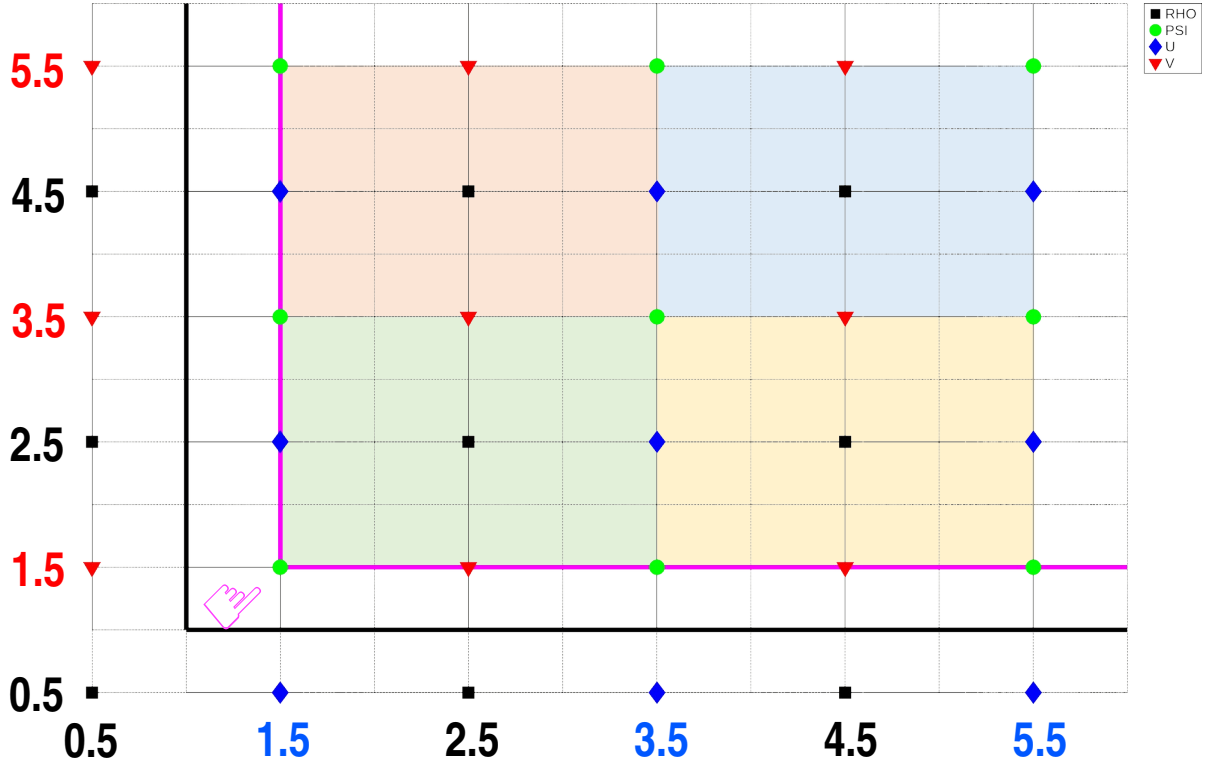
A ROMS grid with dimensions x = 0:L, y = 0:M, and z = 1:N can be decimated by a Gfactor only if MOD(L, Gfactor) = 0 and MOD(M, Gfactor) = 0. Thus, both L and M must be exact multiples of Gfactor.

USEC 3km Grid: 561 x 243 x 50, L = 560, M = 242, N = 50

USEC 6km Grid: 281 x 122 x 50, L = 280, M = 121, N = 50



USEC 3km Grid



Domain Southwestern Corner

USEC 6km Grid

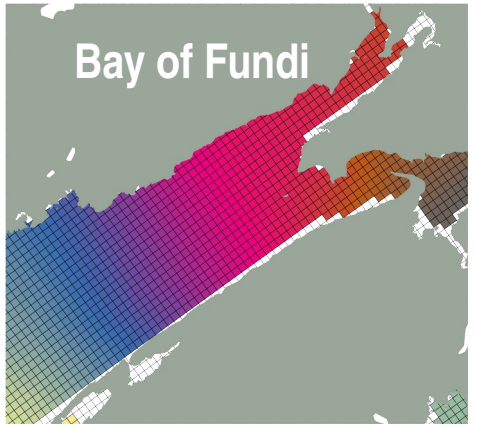
Grid Decimation: Reduction by a Factor of Two

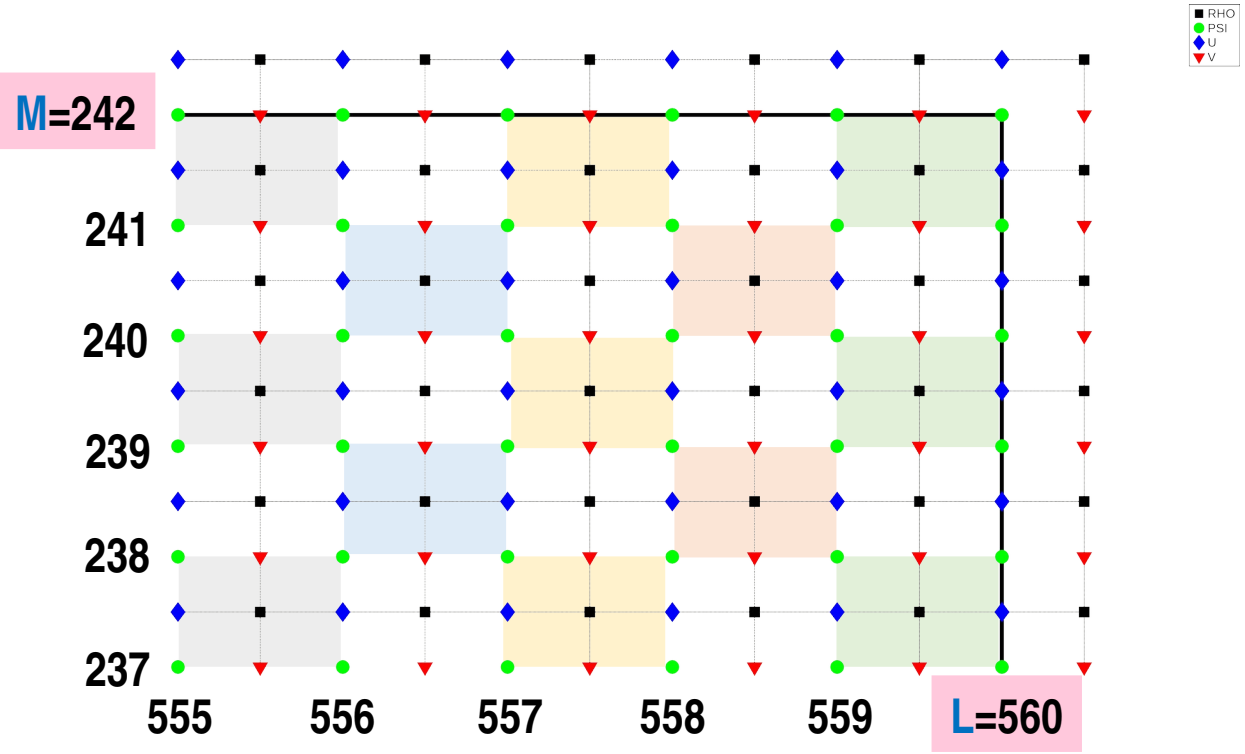
USEC 3km Grid: 561 x 243 x 50, L = 560, M = 242, N = 50

USEC 6km Grid: 281 x 122 x 50, L = 280, M = 121, N = 50

MOD(L, Gfactor) = 0 , Gfactor = 2

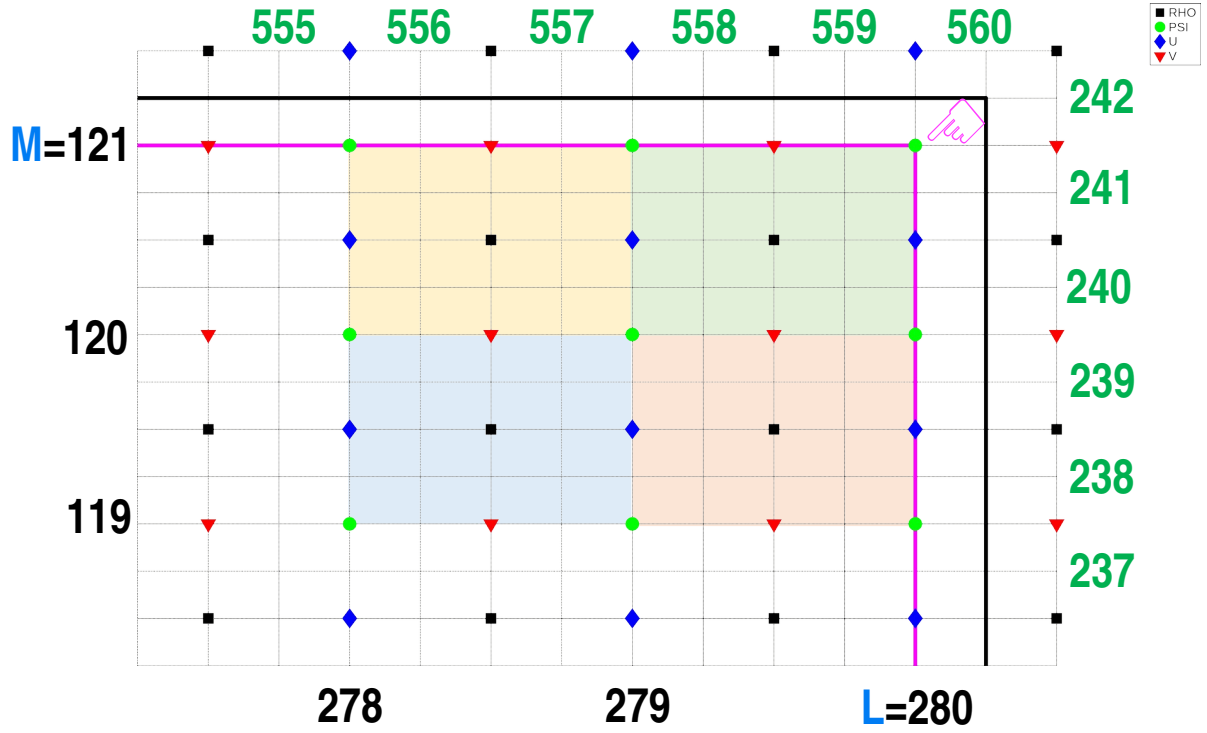
MOD(M, Gfactor) = 0





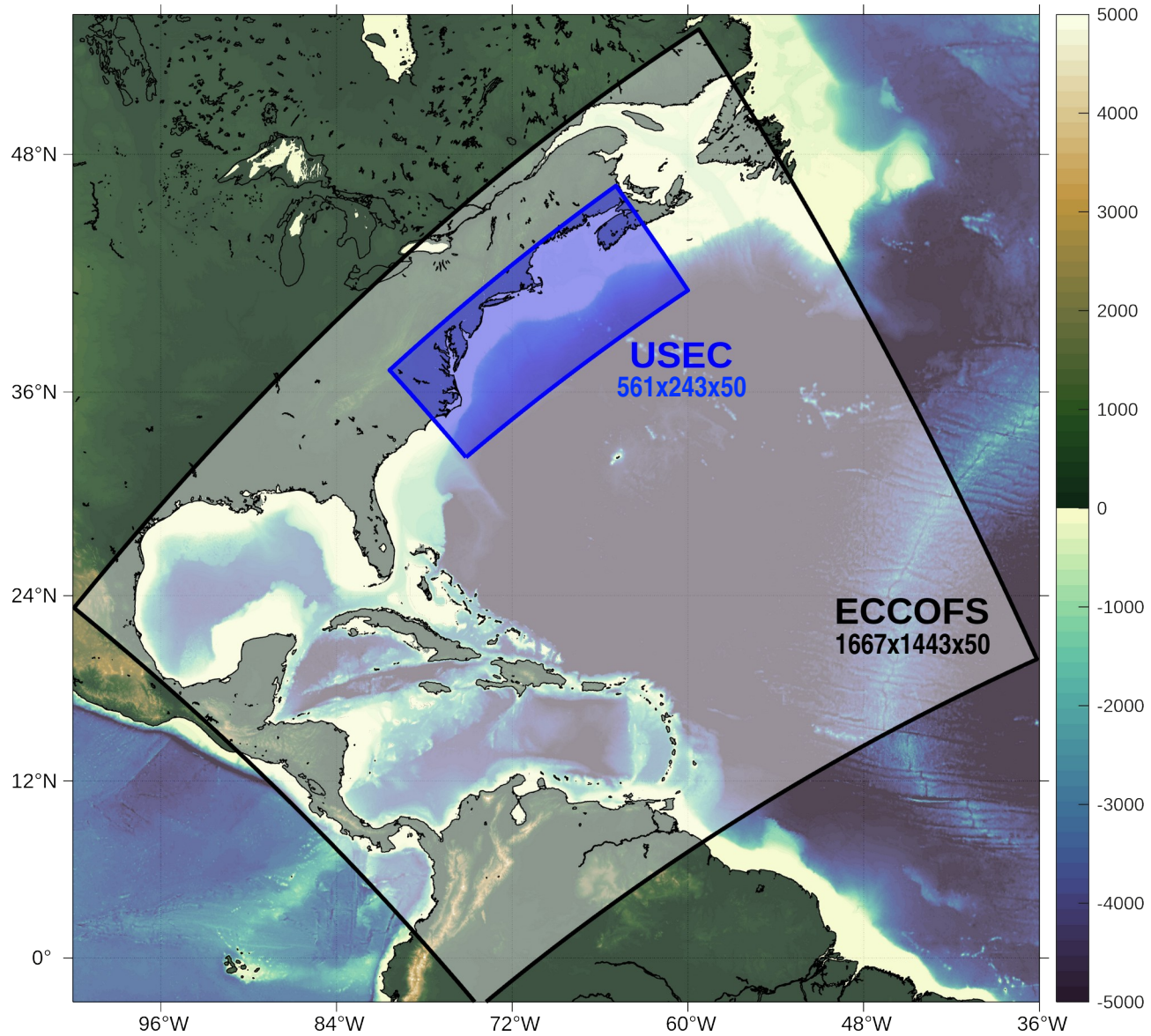
USEC 3km Grid

Domain Northeastern Corner

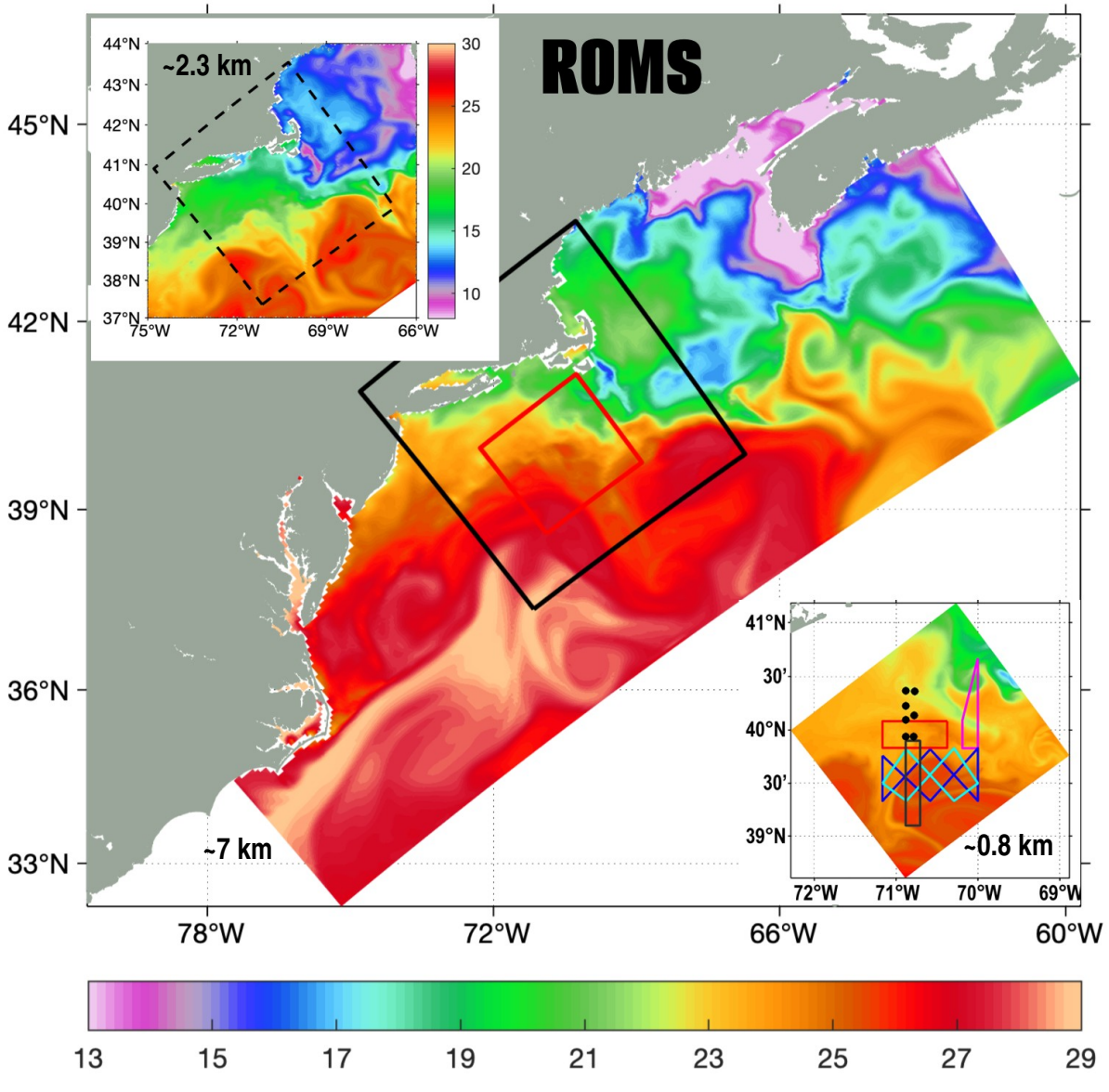


USEC 6km Grid

U.S. East Coast Grids

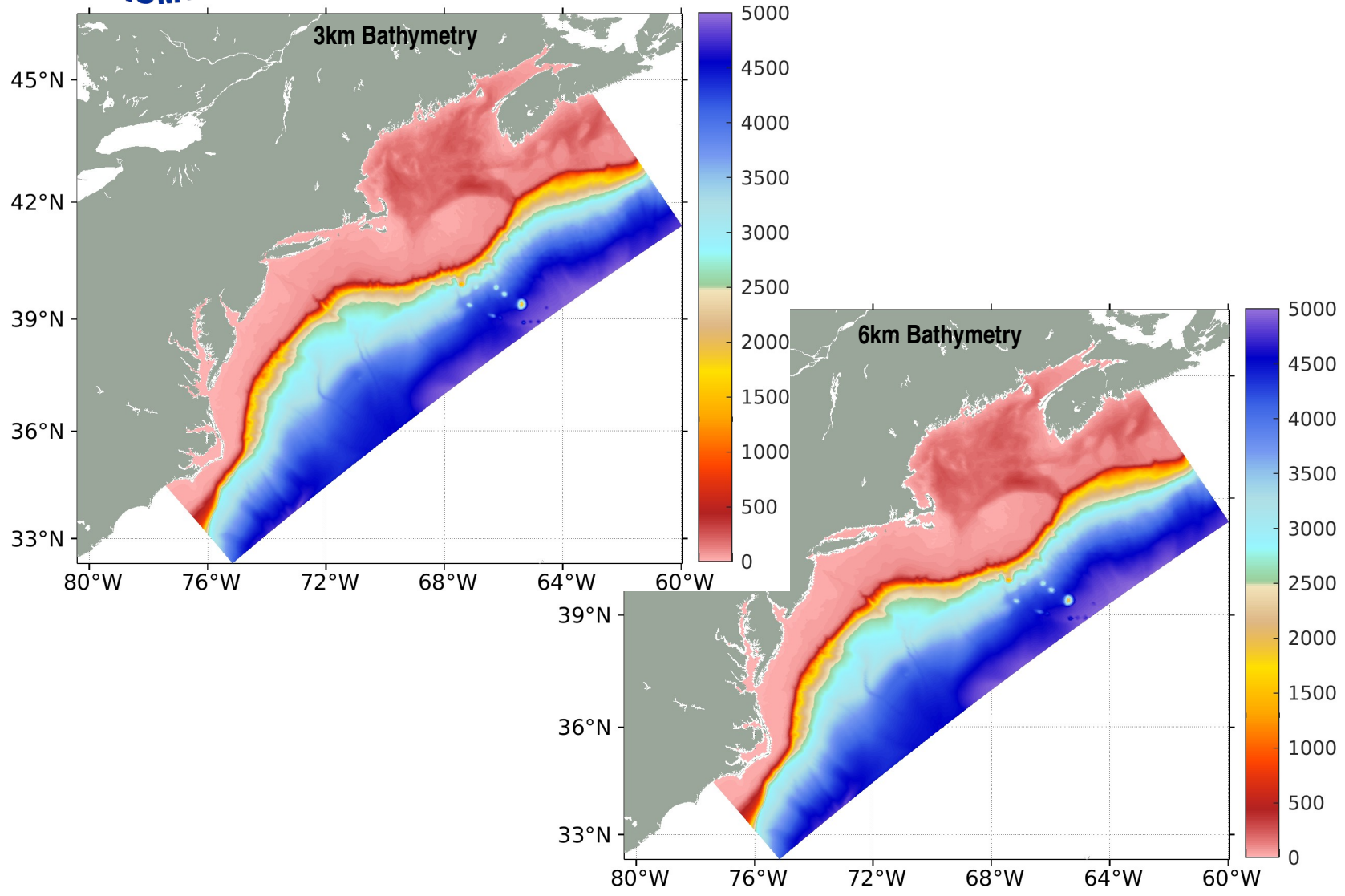


Operational Split 4D-Var (3km/6km)



Triple Nested 4D-Var (1:3 ratio)

U.S. East Coast Regional Application: USEC 3km and 6km Grids



Grid	3km Resolution	6km Resolution
L, M, N	560x242x50	280x121x50
ExtractFlag	2	N/A
DT	120 s	240 s
Unique CPPs	GRID_EXTRACT SPLIT_EXECUTABLE	N/A
Precision	Double/Single	Double/Single
Input and Output Files	PIO	PIO

The GRID_EXTRACT decimates the NLM trajectory, which linearizes the TLM and ADM kernels in the coarser inner loops.

The SPLIT_EXECUTABLE is needed to manage ROMS memory better so large applications may fit in the computer.

U.S. East Coast Regional Application: USEC Split 4D-Var Study

			Mixed Resolution		Mixed Precision			
4D-Var Cases	Outer Loops Iterations	Inner Loops Iterations	Outer Loops Resolution	Inner Loops Resolution	Outer Loops Precision	Inner Loops Precision	Elapsed Time 12 CPUs	Efficiency Percentage Gain
1	1	16	3 km	3 km	Single	Single	9:21:04	58.64%
2	1	16	3 km	3 km	Double	Double	20:24:16	9.74%
3	1	16	3 km	3 km	Double	Single	17:49:16	21.17%
4	2	8	3 km	3 km	Single	Single	9:5936	55.79%
5	2	8	3 km	3 km	Double	Double	22:36:27	Control
6	1	16	3 km	6 km	Double	Single	1:56:17	91.42%
7	1	16	3 km	6 km	Single	Single	1:31:27	93.26%
8	1	16	3 km	6 km	Double	Double	2:36:07	88.49%
9	2	8	3 km	6 km	Double	Single	2:27:44	89.11%
10	1	32	3 km	6 km	Double	Single	3:01:49	86.59%

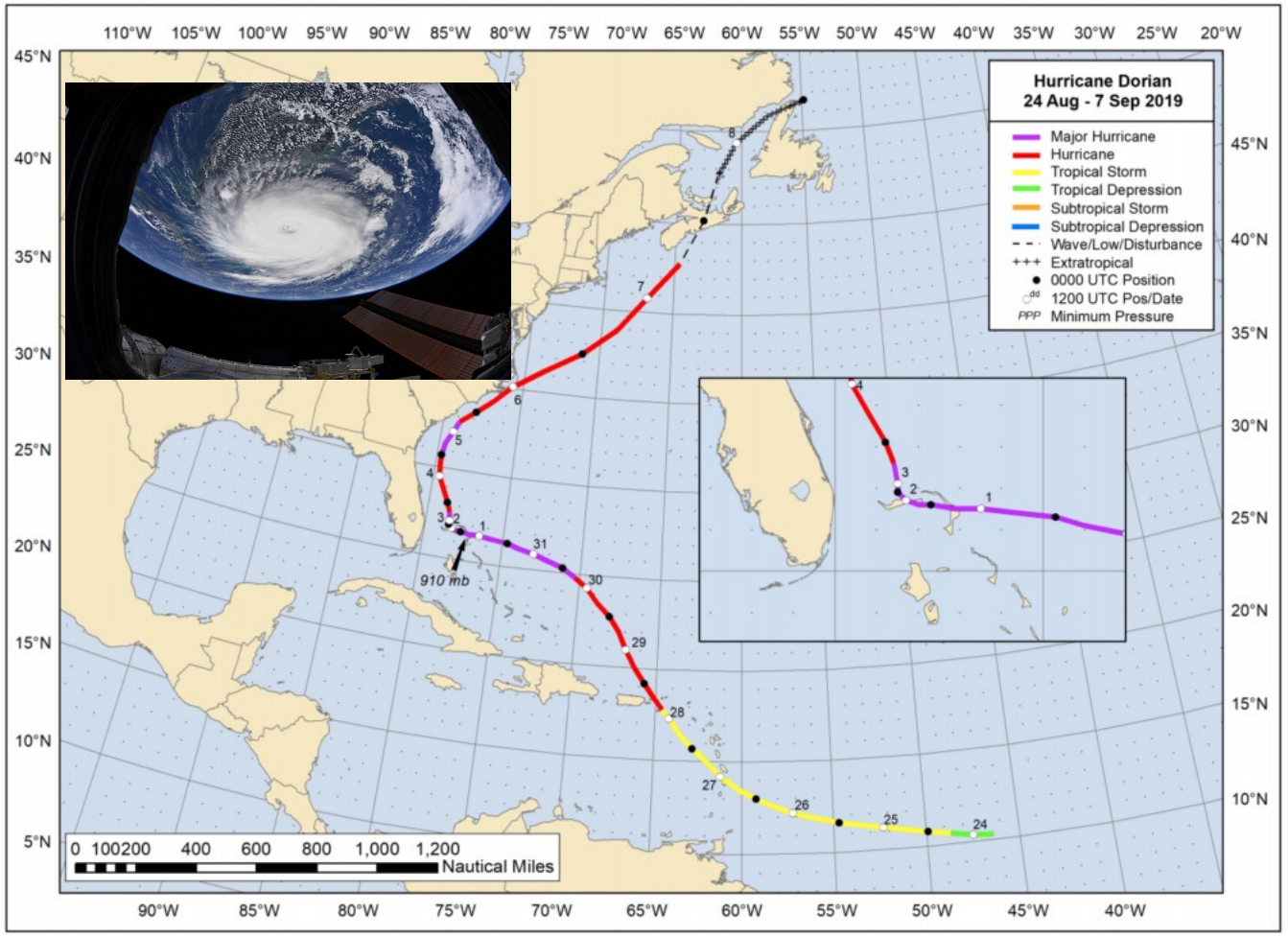
Single-precision inner loops are possible in some cases, but not recommended, as they can affect the stability of the tangent linear and adjoint trajectories

☜ Advise

Mixed Resolution and Mixed Precision require 4D-Var Split Algorithms

U.S. East Coast Regional Application: Hurricane Dorian

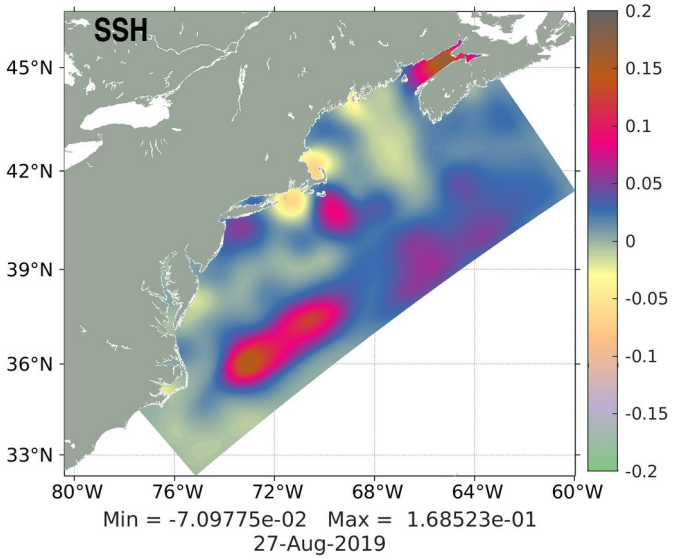
Dorian was a powerful, catastrophic Category 5 hurricane that made landfall in the Bahamas on September 1, 2019, with maximum sustained winds of 185 mph (295km/h).

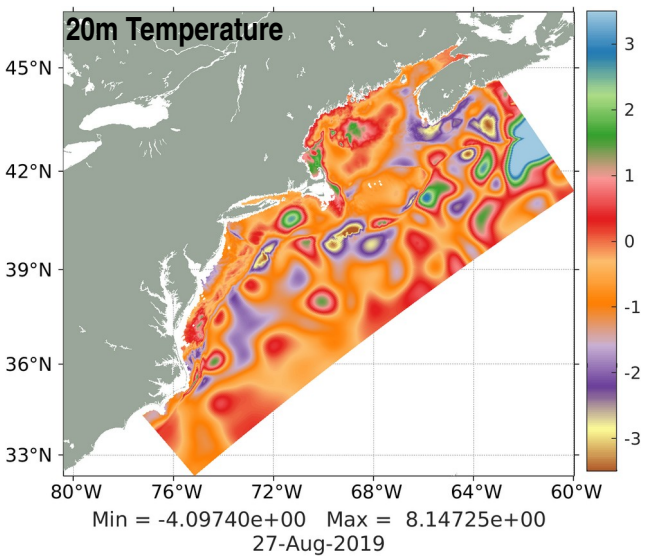


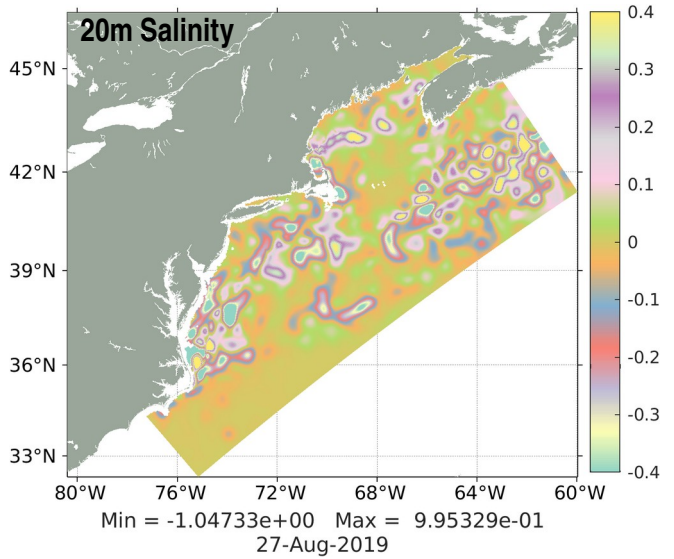
Observations
IR SST: ACSPO (NOAA JPSS/VIIRS, METOP AVHRR)
IR SST: GOES-16 ABI L3C (2km hourly)
Microwave SST: AMSR2 L3U
Altimetry: Sentinel, Jason-3, AltiKa, CryoSat, SWOT
In situ T, S: Argo, gliders, drifters, vessel CTD
eMOLT Fishing Fleet bottom temperatures
HF Radar: CODAR

USEC Mixed-Resolution 4D-Var

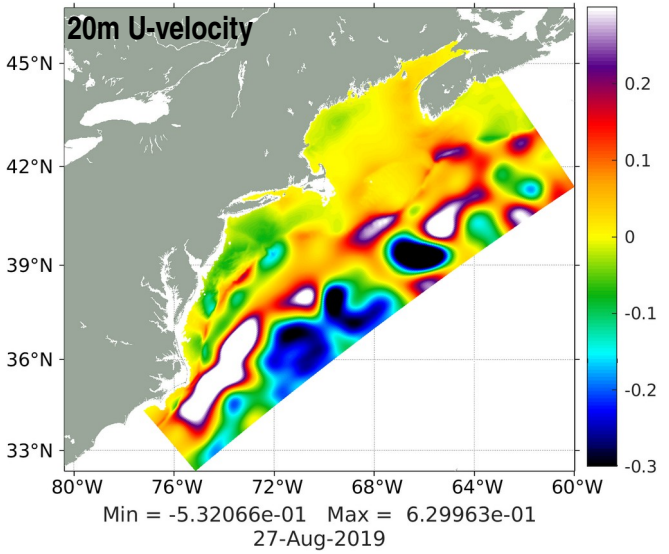
3km Increments

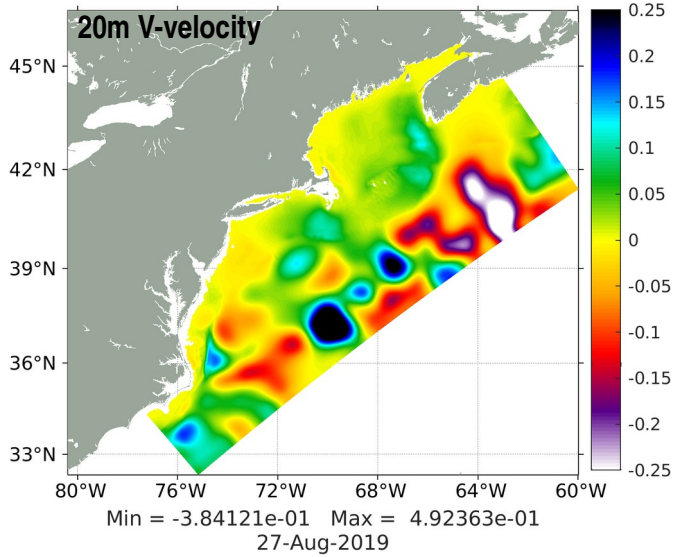




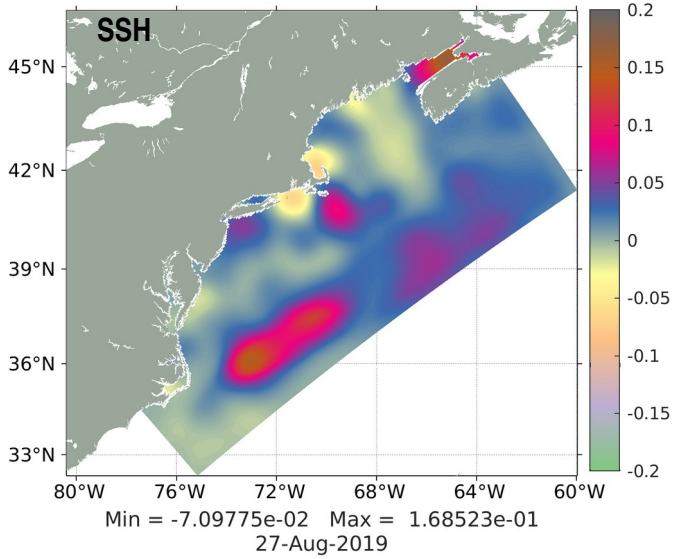


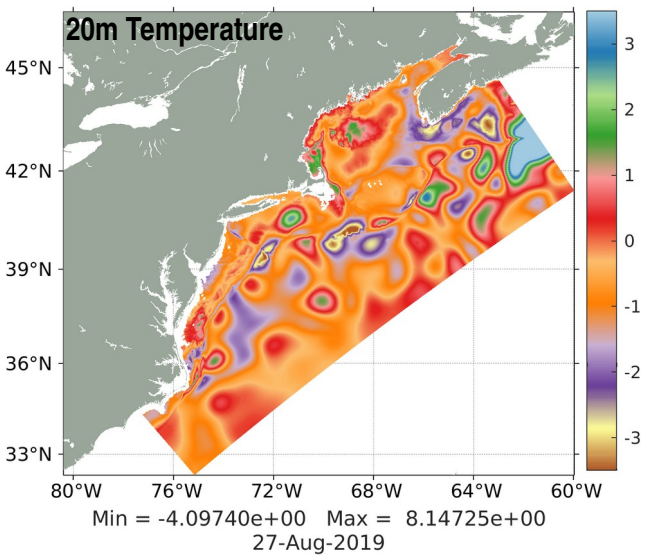
4DVAR Case 8	Outer Loops	Inner Loops
Iterations	1	16
Resolution	3 km	6 km
Precision	Double	Double

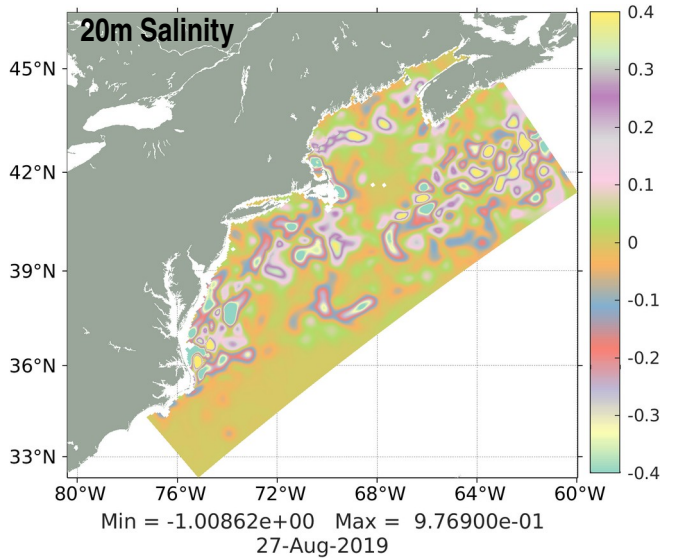




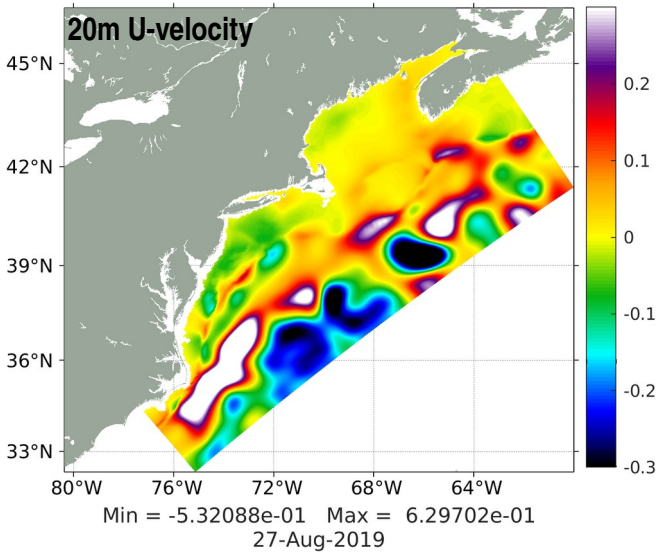
USEC Mixed-Resolution 4D-Var
6km Increments

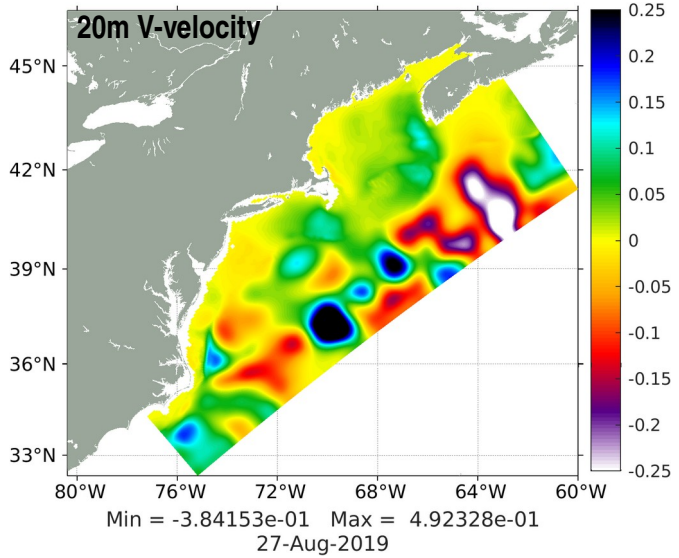






4DVAR Case 8	Outer Loops	Inner Loops
Iterations	1	16
Resolution	3 km	6 km
Precision	Double	Double

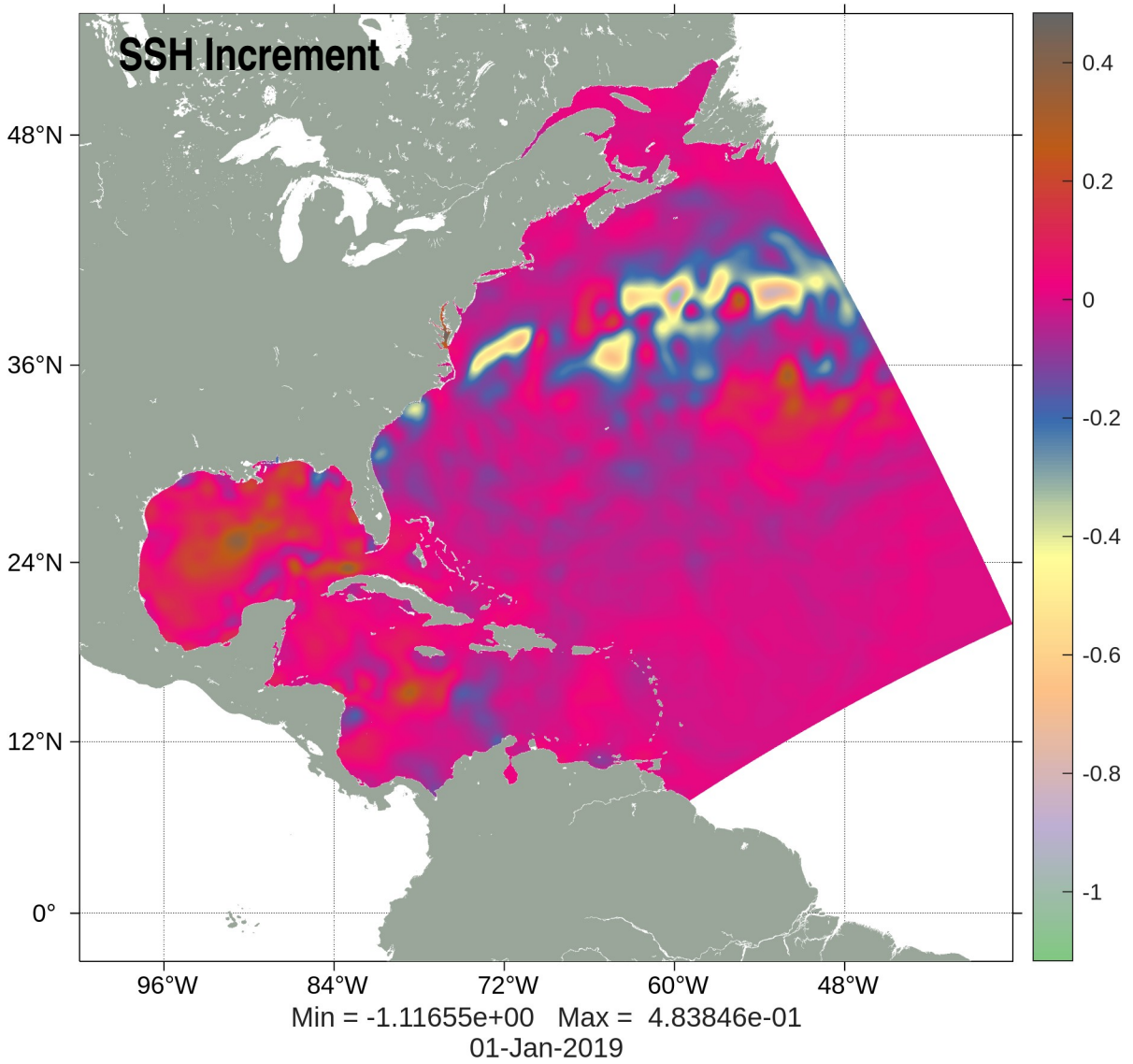


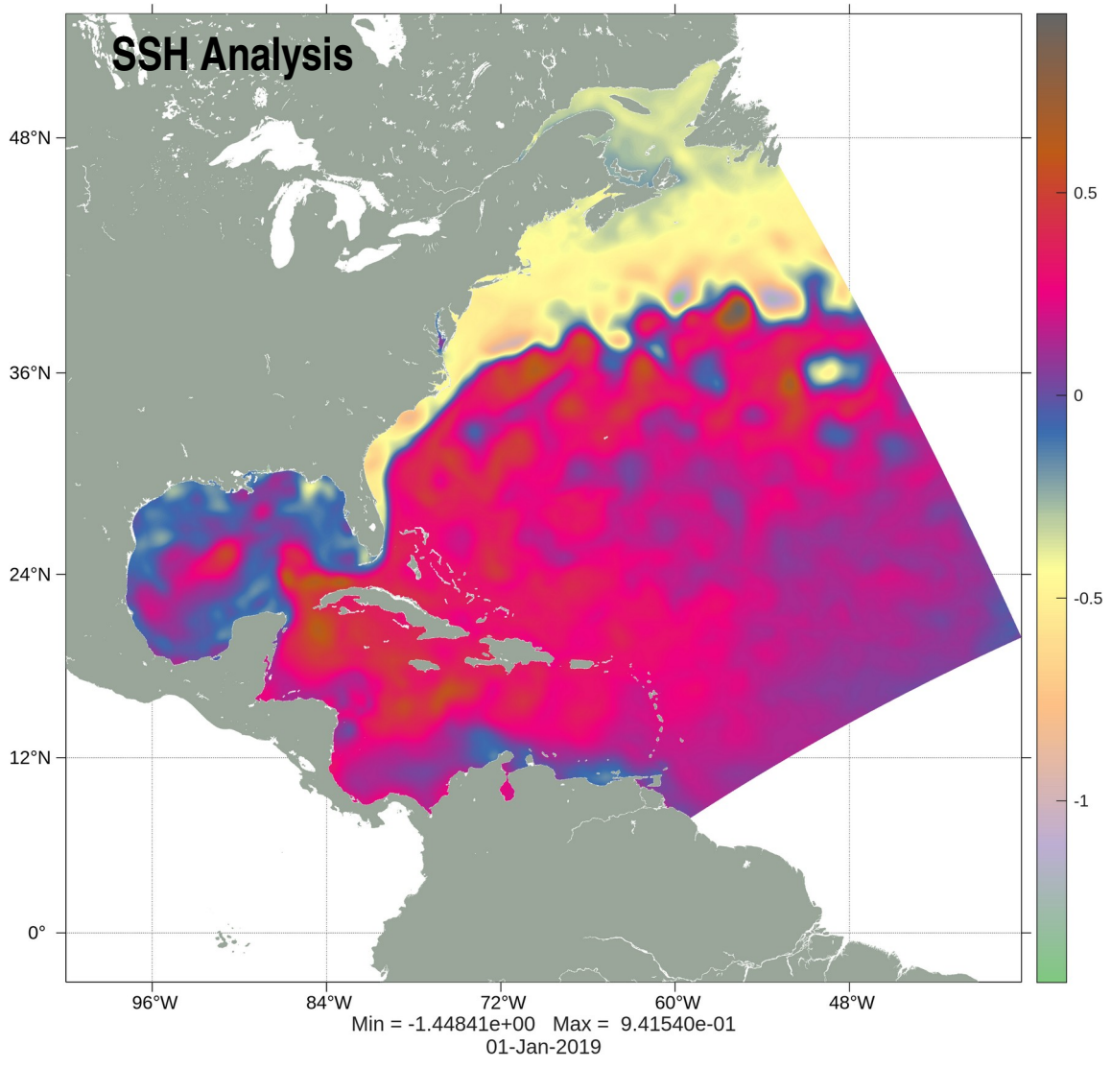


ECCOFS Mixed-Resolution 4D-Var

3km Increment and Analysis

github.com/myroms/roms_eccofs

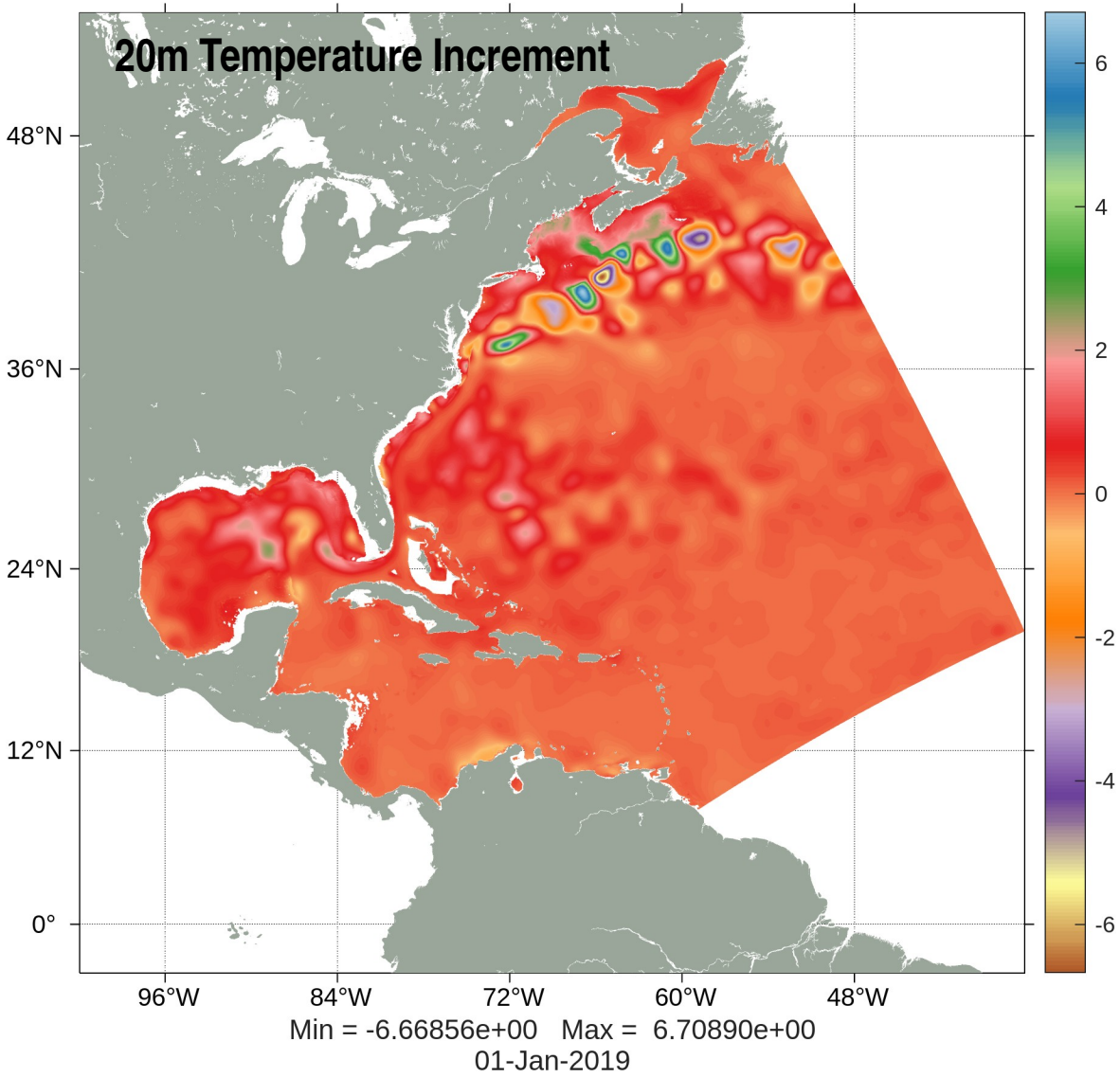


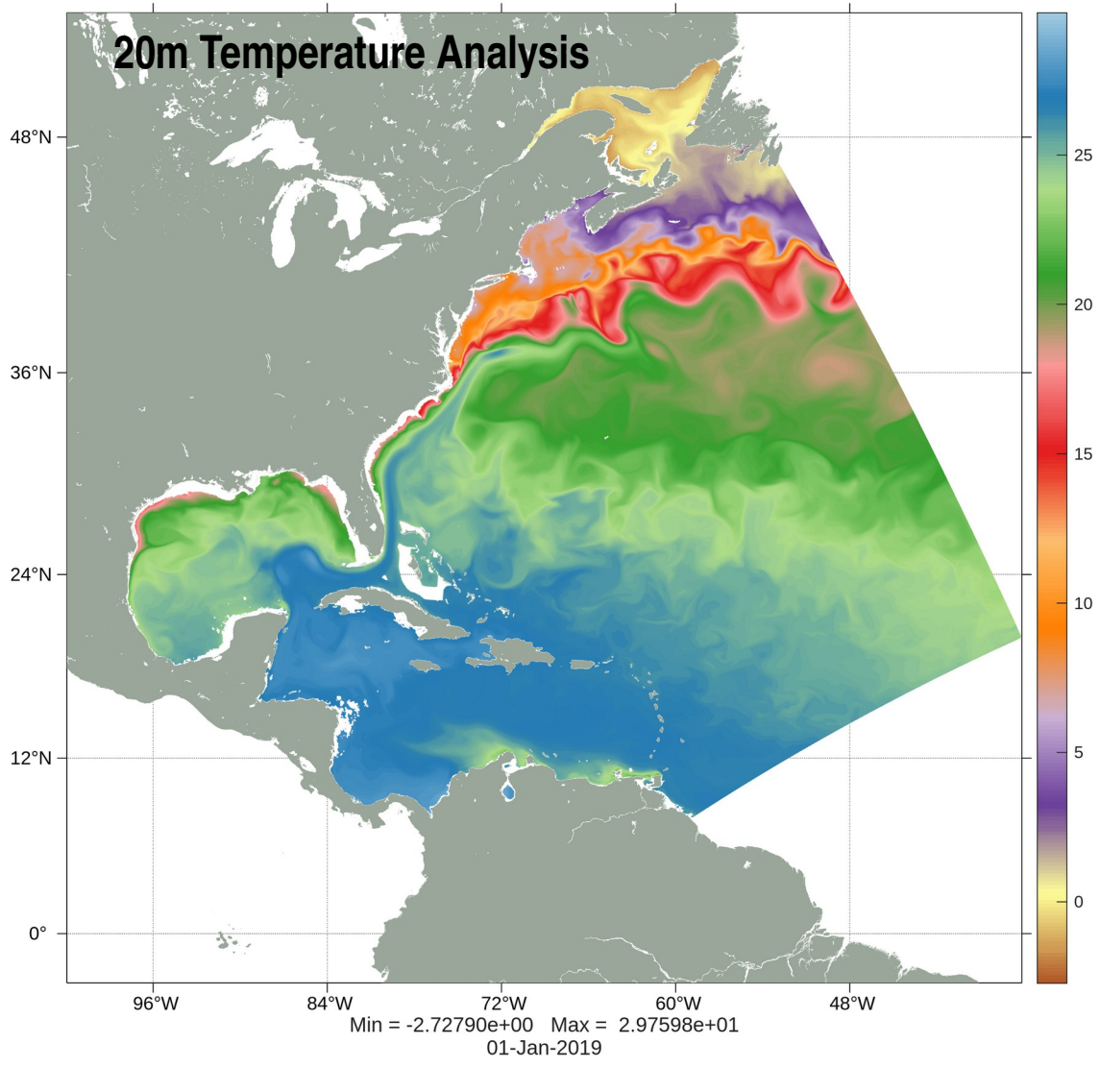


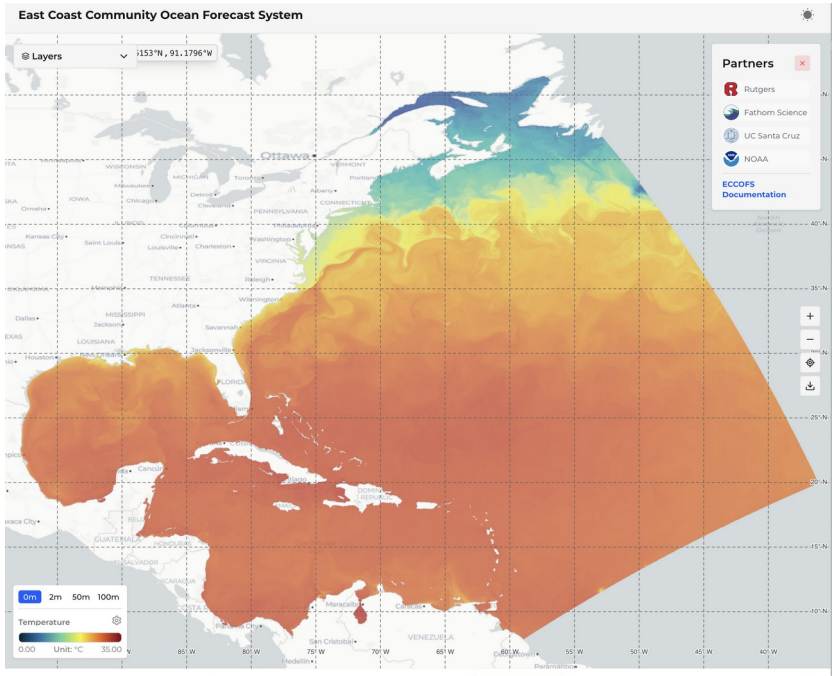
ECCOFS Mixed-Resolution 4D-Var

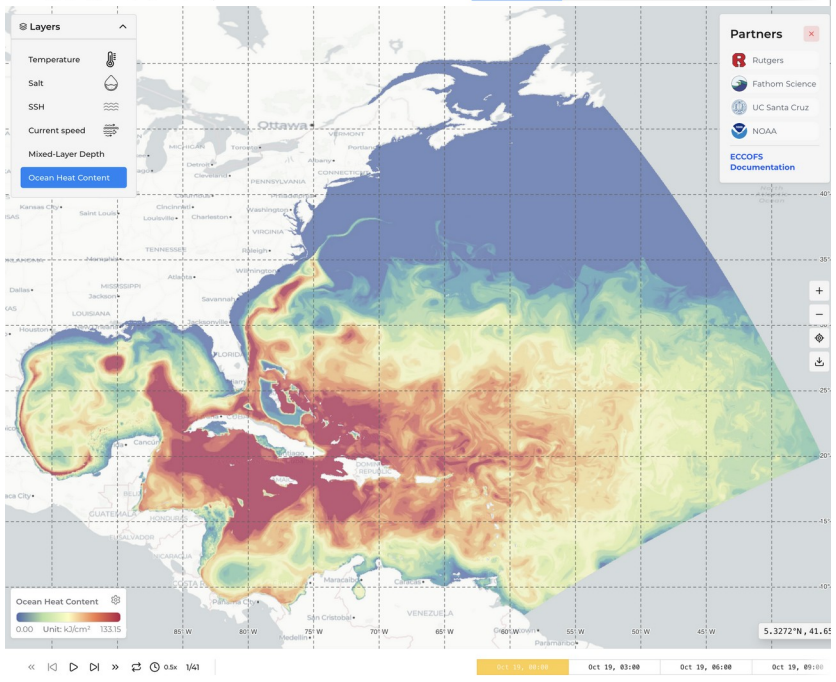
3km Increment and Analysis

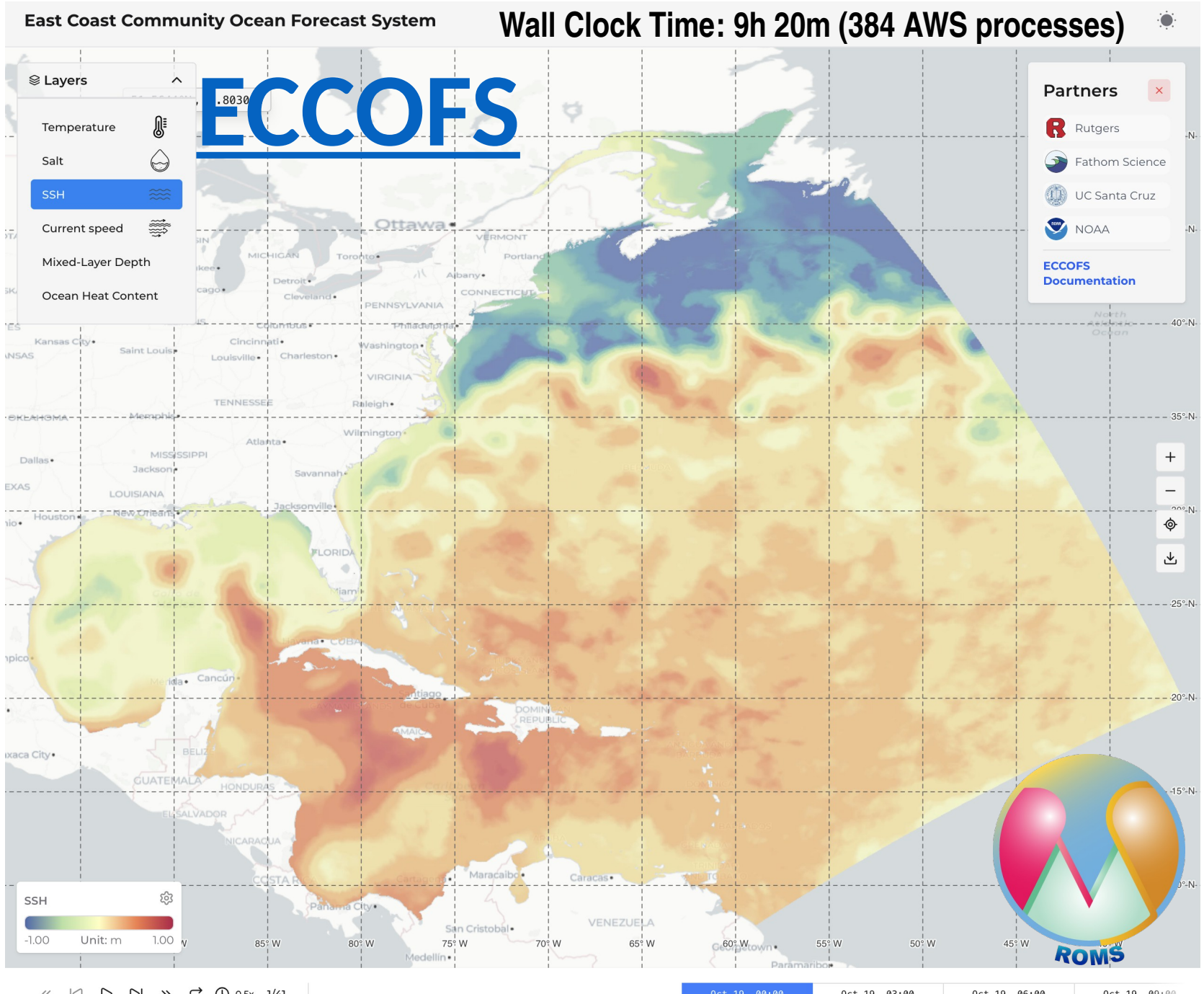
github.com/myroms/roms_eccofs






East Coast Community Ocean Forecast System
Layers
Partners
Rutgers
Fathom Science
UC Santa Cruz
NOAA
ECCOFS Documentation
0m 2m 50m 100m
Temperature
0.00 Unit: °C 35.00


Layers
Temperature
Salt
SSH
Current speed
Mixed-Layer Depth
Ocean Heat Content
Partners
Rutgers
Fathom Science
UC Santa Cruz
NOAA
ECCOFS Documentation
Ocean Heat Content
0.00 Unit: kJ/cm² 133.15


East Coast Community Ocean Forecast System
Wall Clock Time: 9h 20m (384 AWS processes)
ECCOFS
Layers
Temperature
Salt
SSH
Current speed
Mixed-Layer Depth
Ocean Heat Content
Partners
Rutgers
Fathom Science
UC Santa Cruz
NOAA
ECCOFS Documentation
SSH
-1.00 Unit: m 1.00
ROMS
0.5x 1/41
Oct 19, 00:00
Oct 19, 03:00
Oct 19, 06:00
Oct 19, 09:00

Multi-scale Background-Error Covariance

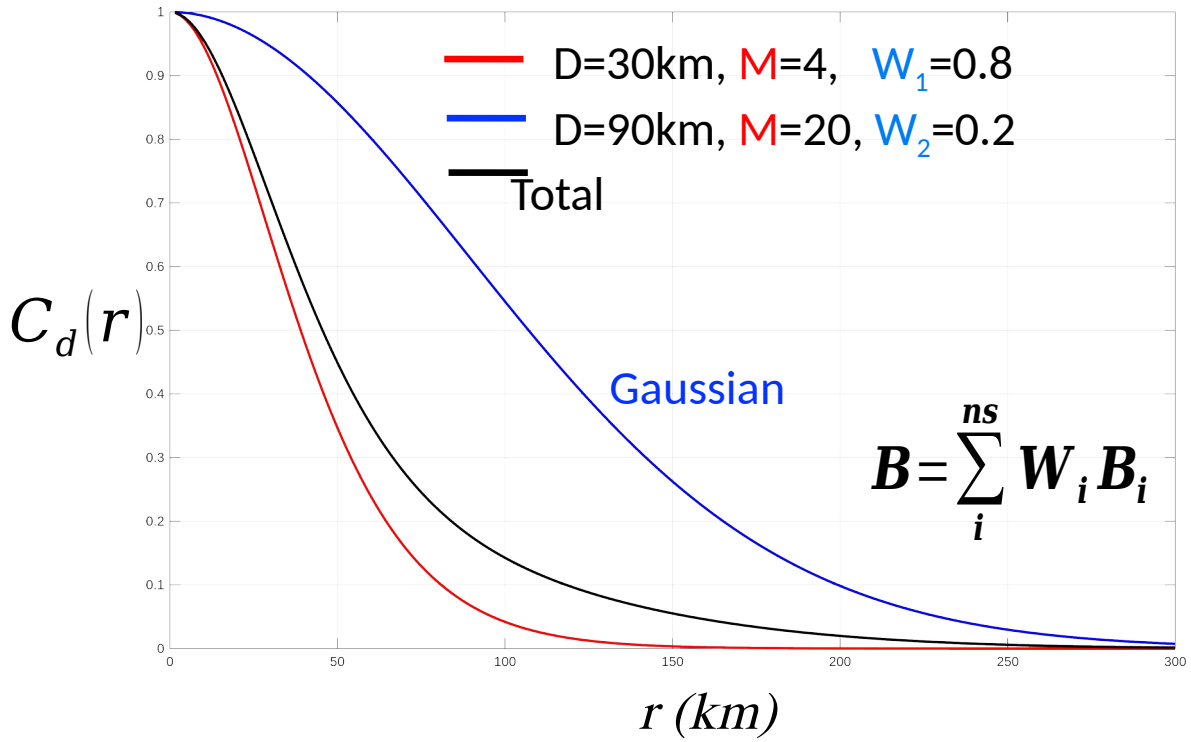
The B-matrix is factorized according to Weaver and Courtier (2001) as

Spreading is modeled with pseudo-diffusion operators in spatial correlation space (Weaver *et al.*, 2016, 2018):

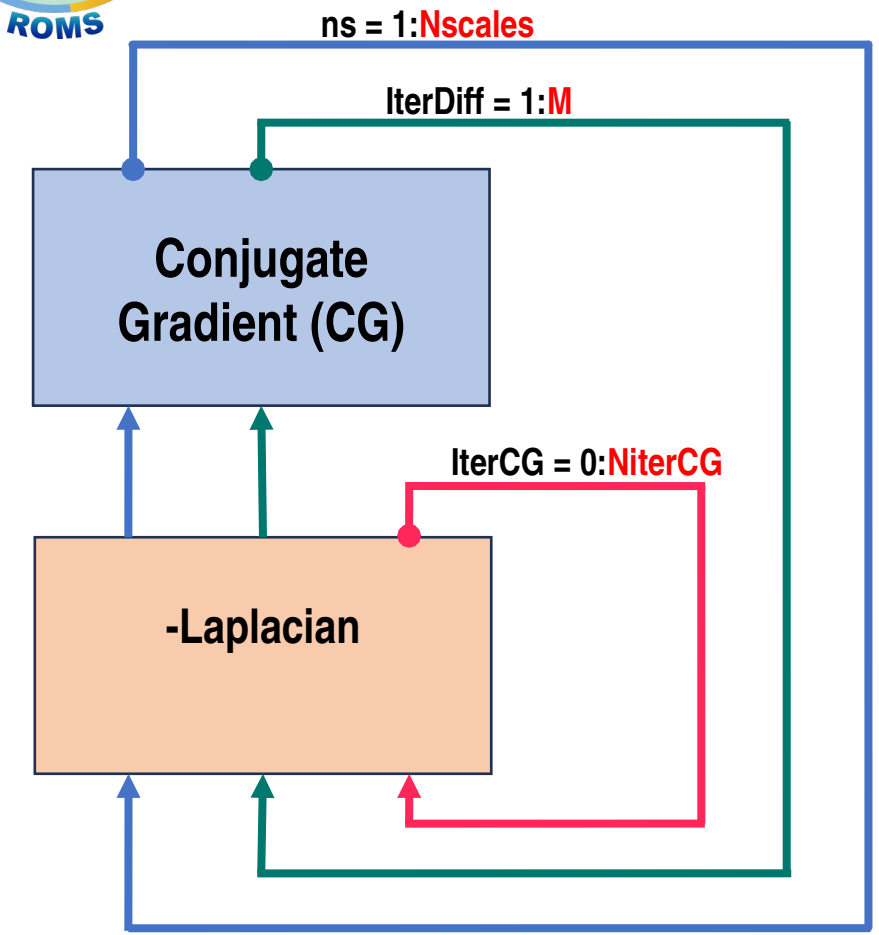
is a diffusion coefficient proportional to the correlation length squared, , (Daley, 1992). Implicit solutions encompass a wider class of Matérn correlation functions (Weaver and Mirouze , 2013):

The discretized implicit 2D diffusion equation after iterations is expressed as:

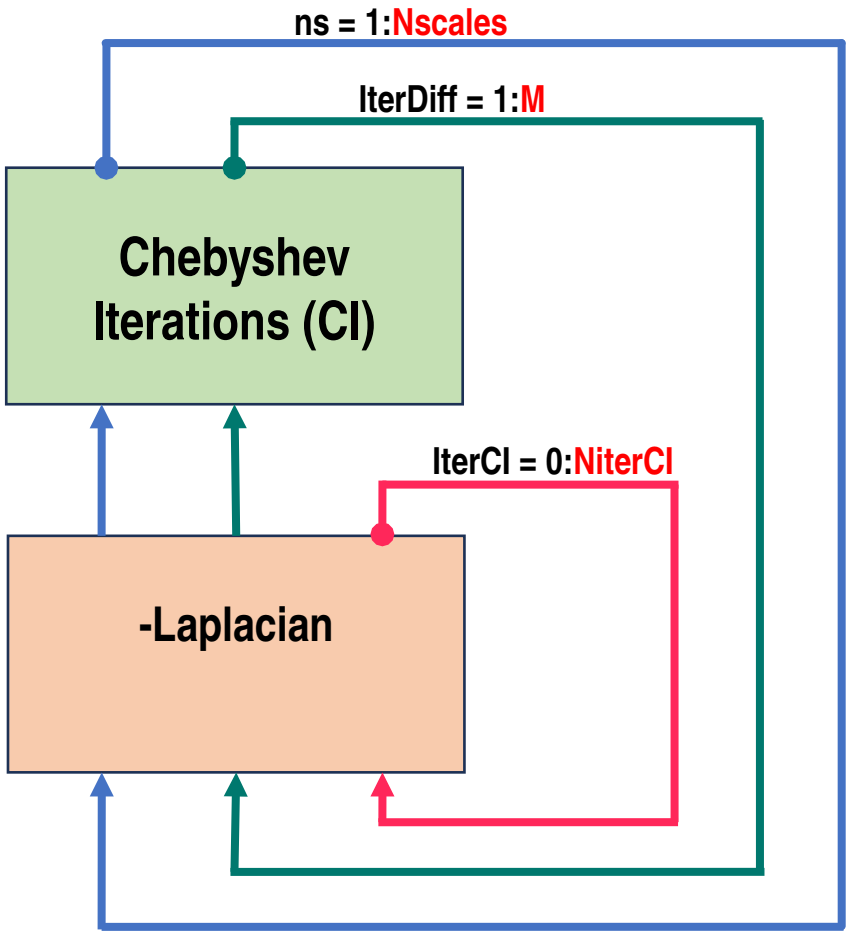
Solutions are related to the Matérn function by
order of Bessel function
for 2D case,
Daley (1992) length scale
isotropic,
anisotropic tensor



Implicit Multi-scale Diffusion Operator
(Weaver et al., 2016, Algorithm 3)



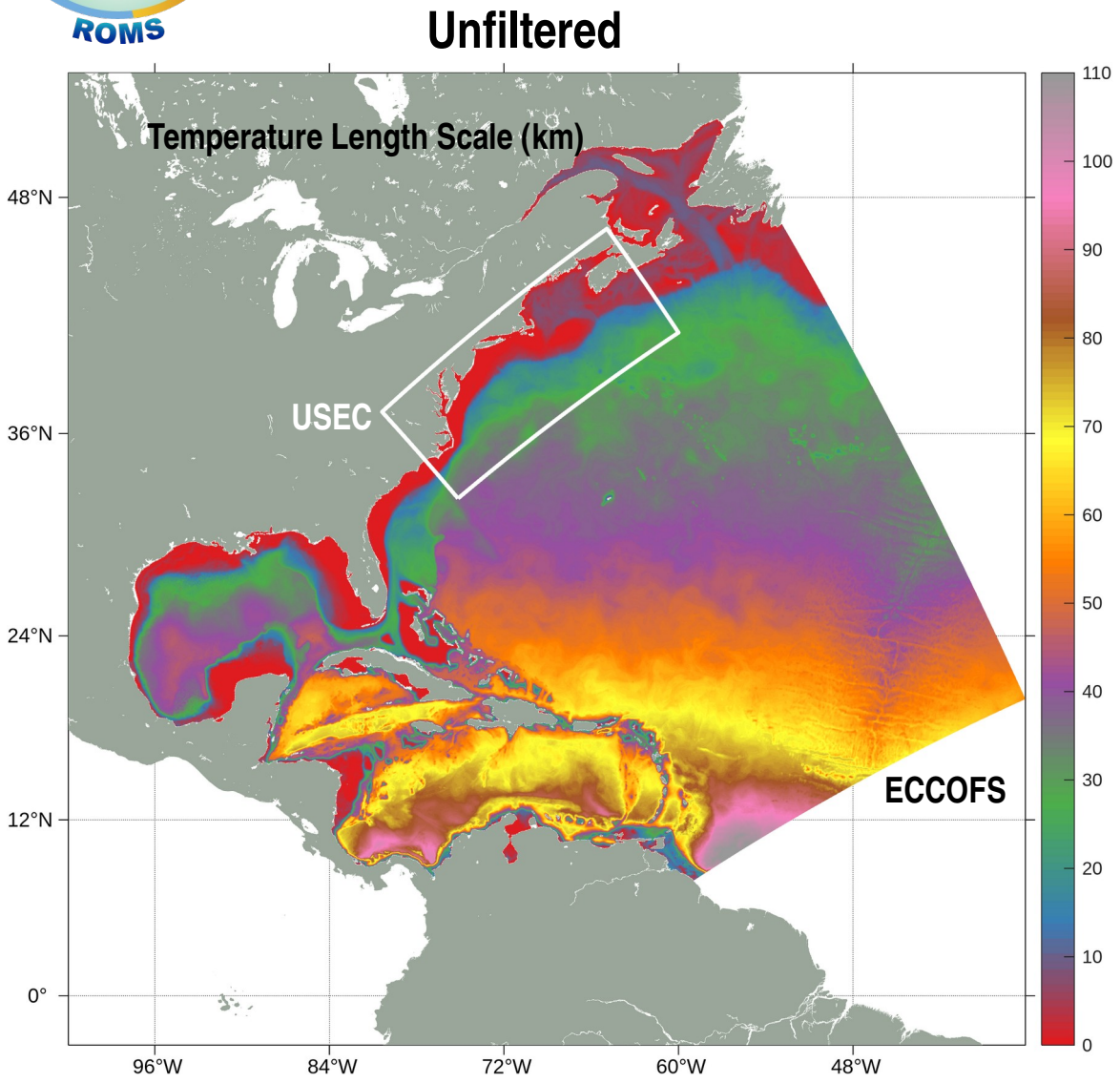
Modeling the spreading and smoothing effects of $\mathbf{B}_i$

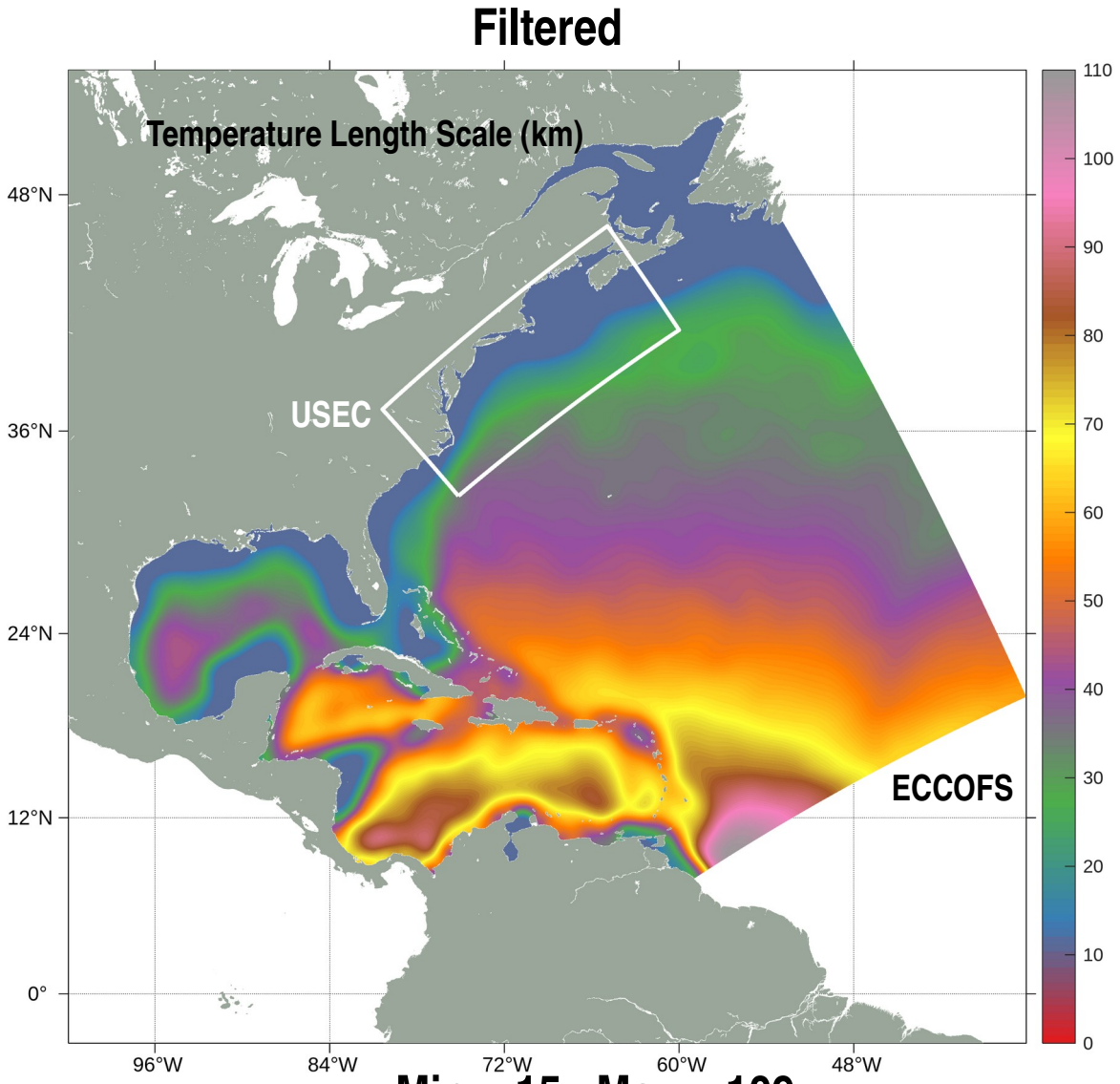


Computes the extreme Ritz eigenvalues ($\mathbf{Ax} = \mathbf{b}$) required by the Chebyshev Iterations (CI) Solver (initialized with random vectors)

Inverts the implicit Pseudo-Diffusion Equation (eigen spectrum is invariant for fixed grid and

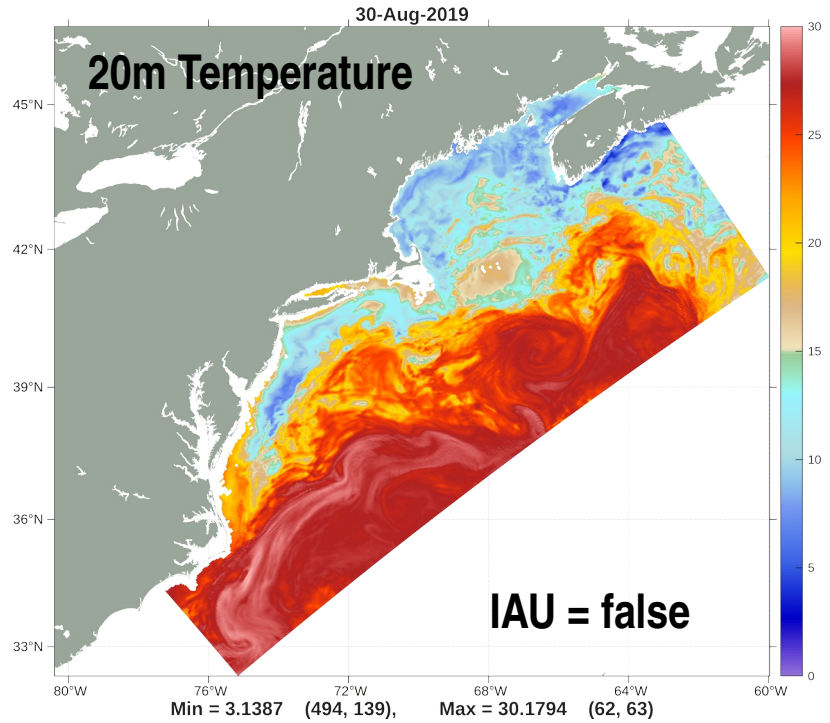
Spatially-Varying Correlations
(Based on the First Rossby Radius: L_x and L_y)





Mono-scale versus Multi-scale: Next DA cycle IC (Explicit versus Implicit)

Mono-scale B



Min = 3.1387 (494, 139), Max = 30.1794 (62, 63)

Isotropic

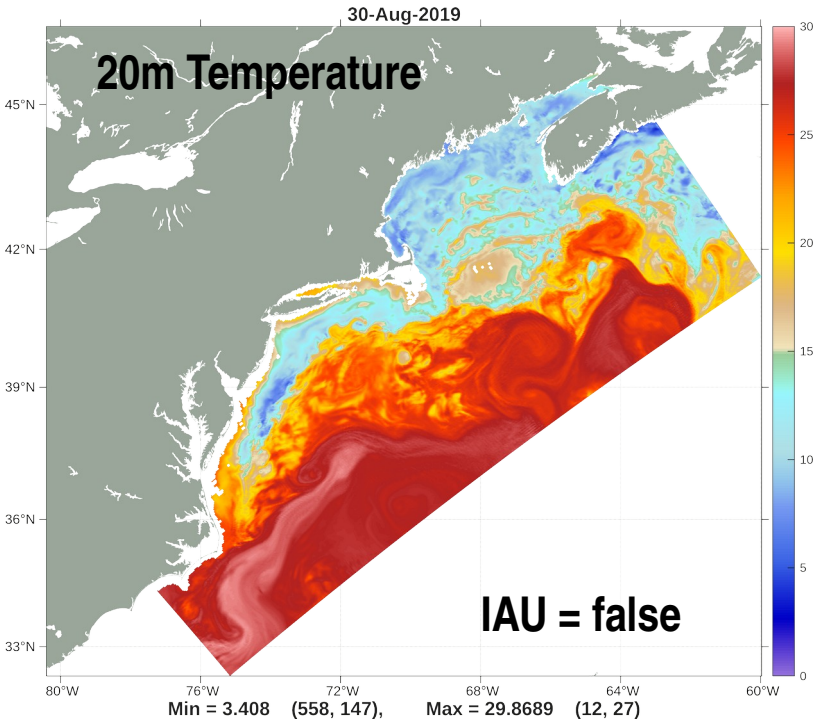
L_H = 40 km

Explicit

L_V = 10 m

Implicit

Multi-scale B_i



Min = 3.408 (558, 147), Max = 29.8689 (12, 27)

Isotropic, M=10, Nscales=1

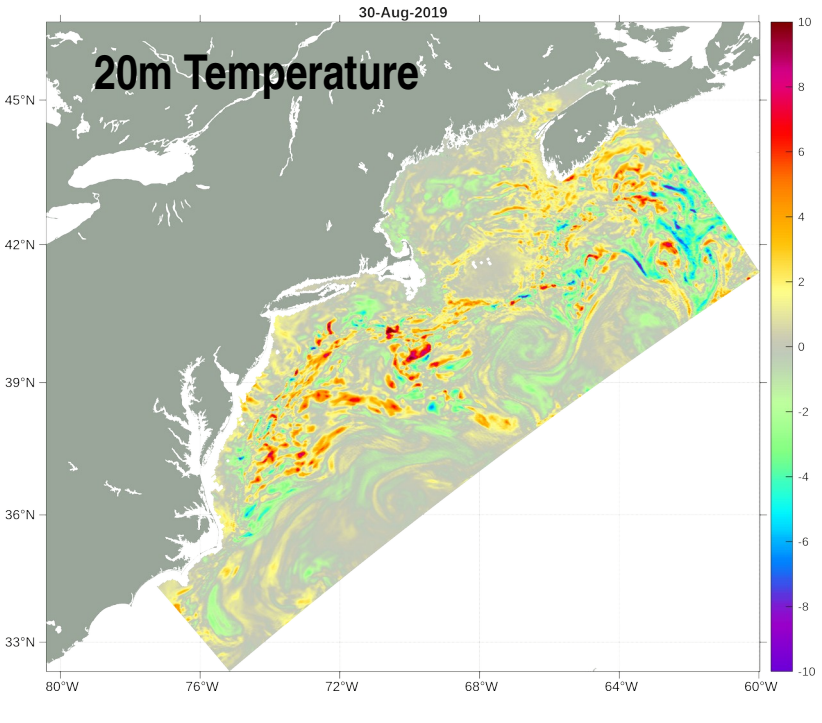
$L_{x,y}$ **Rossby Radius**

Implicit

L_V = 10 m

Implicit

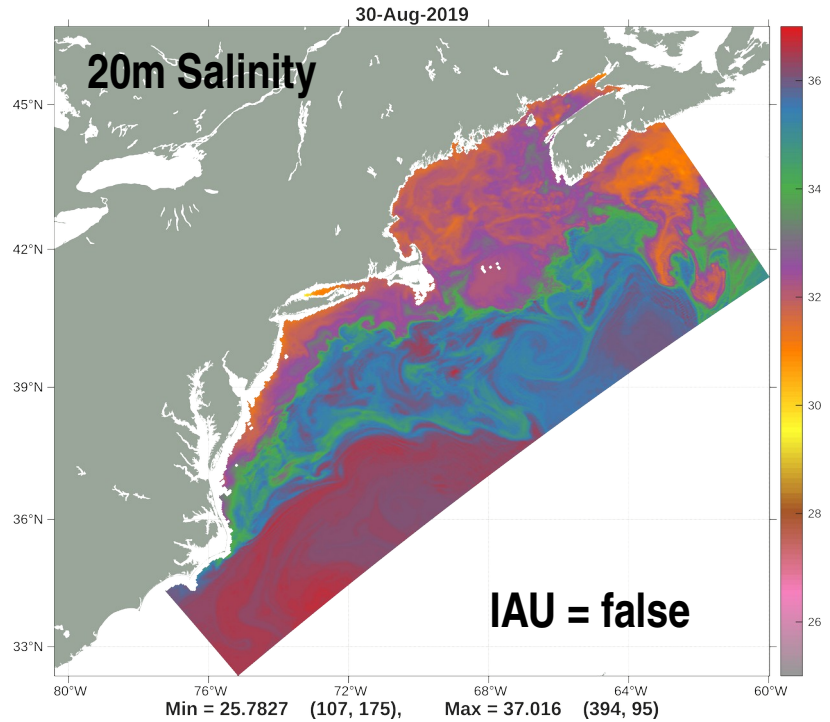
Difference



Multi-scale minus Mono-scale

Mono-scale versus Multi-scale: Next DA cycle IC (Explicit versus Implicit)

Mono-scale B



Isotropic

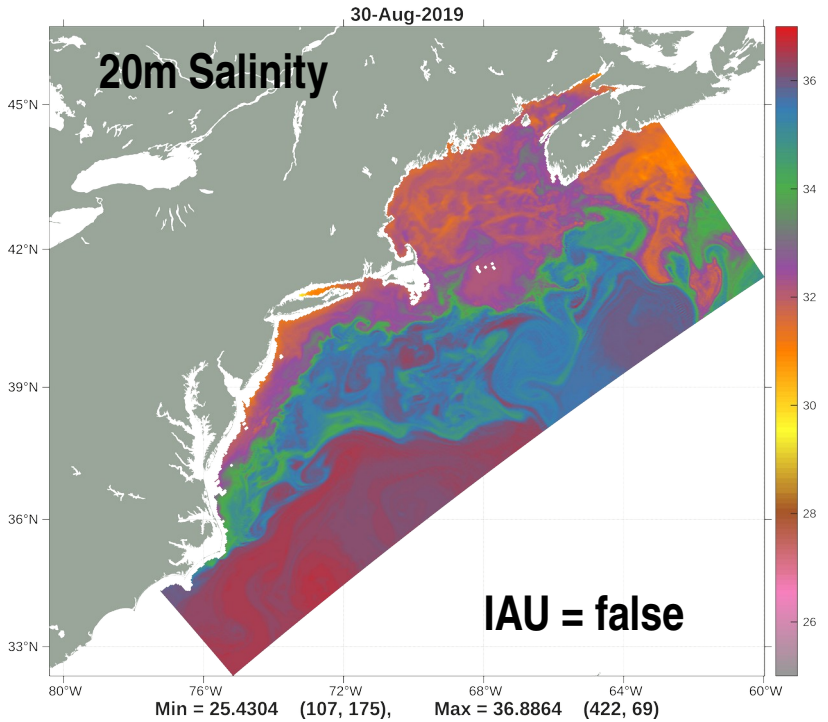
L_H = 15 km

Explicit

L_V = 10 m

Implicit

Multi-scale B_i



Isotropic, M=10, Nscales=1

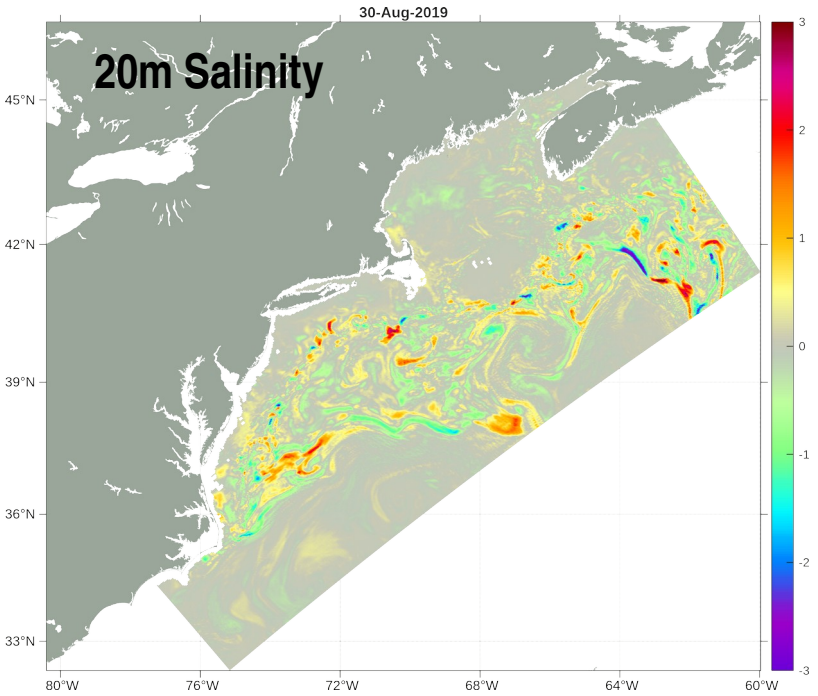
$L_{x,y}$ Rossby Radius

Implicit

L_V = 10 m

Implicit

Difference



Multi-scale minus Mono-scale

Observations and Operators
IODA Version 3, Enhanced NetCDF-4
ROMS and ROMS-JEDI

- The consolidation strategy involves using IODA-type enhanced NetCDF-4 files with first-level grouping in the native ROMS 4D-Var and ROMS-JEDI data assimilation algorithms. A NetCF-4 Group is based on the underlying HDF5 data model. It is equivalent to a subdirectory in a file and allows hierarchical structuring of observation data within a single file.
- Different groups may contain variables with the same name without conflict, allowing storage of various data properties, such as observation values, observation errors, quality control flags, model H(x) values, background errors, innovations, increments, residuals, and many others. Note that IODA-type files are used for input and output in a data assimilation cycle.
- The strategy is to have an individual IODA-type observation file for each variable in the control vector, such as SSH, SST, SSS, HF radar velocities, temperature, and salinity, in a particular data assimilation cycle.
- A file can include multiple variables representing vectors (u- and v-momentum) and spectral data with exact coordinates (lon, lat, depth, time). That is, the observations are stored by location.
- It may also contain multiple scalar variables, such as T and S profiles or biological profiles for various ecosystem state variables, at the precise location (lon, lat, depth, time).
- It may also include secondary variables to compute a control variable with an elaborate operator. This structure provides us with considerable flexibility for filtering and space/time averaging operators.
- For more information, please check: github.com/myroms/roms/wiki/ROMS-Observation-Operators.

Final Remarks

- The mixed-resolution 4D-Var is highly efficient and significantly reduces computational wall time in demanding, high-resolution operational applications. This approach is over 85 percent faster than executing both the outer and inner loops at the same fine resolution.
- The mixed-precision configuration yields efficiency gains of approximately 3 percent; however, it is not recommended because it may compromise the stability of the tangent linear and adjoint kernels.
- Grid decimation serves as an effective strategy for reducing grid resolution in the inner loops. Extrapolation during fine-to-coarse and coarse-to-fine operations is avoided. However, challenges in land/sea masking between fine and coarse grids constrain the decimation factor to Gfactor = 2.
- The multi-scale background error covariance operator described by Weaver *et al.* (2016, 2018) and Mirouze *et al.* (2016) has been implemented in native ROMS 4D-Var.
- The multi-scale background error formulation would represent both broad structures and fine-scale features, reducing scale aliasing in the data assimilation cost function.
- A key advantage of the implicit horizontal multi-scale operator is its capacity to accommodate spatially varying correlation length scales within the κ-diffusion coefficient tensor.
- Determining the optimal horizontally varying correlation scales for operational 4D-Var applications will require further experimentation.
- The native ROMS observation and H(x) operators have been refactored to support multiple IODA-type, enhanced NetCDF-4 files, enabling area and time averaging for H(x) and $H^T(x)$ processing.

Thank you!

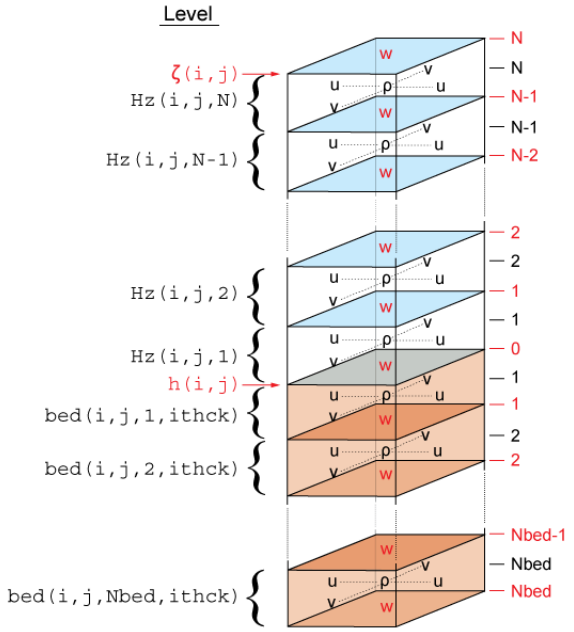
ROMS and ROMS-JEDI Resources

- The ROMS Framework is distributed at:
  https://github.com/myroms/roms
  https://github.com/myroms/roms/wiki
- The ROMS-JEDI interface has a private and a public repository:
  - Private: https://github.com/JCSDA-internal/roms-jedi
  - Public: https://github.com/myroms/roms-jedi
- Each repository includes a GitHub Wiki for documentation and instructions:
  - Private: https://github.com/JCSDA-internal/roms-jedi/wiki
  - Public: https://github.com/myroms/roms-jedi/wiki
- Tutorial for ROMS-JEDI Beginners and Advanced Users:
  https://github.com/myroms/roms-jedi/wiki/ROMS-JEDI-Tutorial
- Documentation and Instructions about IODA Enhanced NetCDF-4 files and Observation Operators:
  https://github.com/myroms/roms/wiki/ROMS-Observation-Operators
- Multiscale Background Error Covariance documentation and instructions:
  https://github.com/myroms/roms_test/tree/main/WC13/Normalization

Additional Material

Observations Metadata

- The table below displays the names of native ROMS 4D-Var and ROMS-JEDI observations in input and output IODA files, along with their relationship to the ROMS control vector. These variables are defined in obsop_name_map.yaml.
- All the observations are assumed to be located at the grid center (A-grid, ρ-point).
- ROMS subsurface temperature observations are in terms of potential temperature.



Short Name	State ID	Standard Name	IODA Variable Name
ssh, SSH	1	sea_surface_height_above_geoid	absoluteDynamicTopography
u, uocn	4	eastward_sea_water_velocity	waterZonalVelocity
v, vocn	5	northward_sea_water_velocity	waterMeridionalVelocity
sst, SST	6, 60	sea_surface_temperature	seaSurfaceTemperature
tocn, temp	6	sea_water_temperature	waterTemperature
sss, SSS	7, 70	sea_surface_salinity	seaSurfaceSalinity
socn, salt	7	sea_water_salinity	salinity

IODA: Enhanced NetCDF-4 Observation File

```
netcdf usec3km_adt_20190827 {
dimensions:
        Location = 1451 ;
        nvars = 1 ;
        survey = 16 ;
variables:
        int Location(Location) ;
                Location:suggested_chunck_dim = 512 ;
        int nvars(nvars) ;
                nvars:suggested_chunck_dim = 100 ;
        int survey(survey) ;
                survey:suggested_chunck_dim = 100 ;
// global attributes:
                :_ioda_layout_version = 3 ;
                :date_time = 2019082700. ;
                :datetimeReference = "2019-08-27T00:00:00Z" ;
group: MetaData {
  variables:
        int64 dateTime(Location) ;
                dateTime:long_name = "elapsed observation time since reference" ;
                dateTime:units = "seconds since 2019-08-27T00:00:00Z" ;
        int64 dateTimeAverageBegin(Location) ;                          // Optional
                dateTimeAverageBegin:long_name = "start of time averaging filter" ;
                dateTimeAverageBegin:units = "seconds since 2019-08-27T00:00:00Z" ;
        int64 dateTimeAverageEnd(Location) ;                            // Optional
                dateTimeAverageEnd:long_name = "end of time averaging filter" ;
                dateTimeAverageEnd:units = "seconds since 2019-08-27T00:00:00Z" ;
        float latitude(Location) ;
                latitude:long_name = "observation latitude" ;
                latitude:units = "degrees_north" ;
        float longitude(Location) ;
                longitude:long_name = "observation longitude" ;
                longitude:units = "degrees_east" ;
        int provenance(Location) ;
                provenance:long_name = "observation origin identifier" ;
        int sequenceNumber(Location) ;
                sequenceNumber:long_name = "observation sequence number" ;
        float spatialAverage(nvars) ;                                   // Optional
                spatialAverage:long_name = "half-length of spatial averaging filter" ;
                spatialAverage:units = "meter" ;
        int stateID(nvars) ;
                stateID:long_name = "state variable index" ;
        int surveyIndex(survey) ;
                surveyIndex:long_name = "survey time indices as they appear in dateTime" ;
        int64 surveyTime(survey) ;
                surveyTime:long_name = "observation survey time" ;
                surveyTime:units = "seconds since 2019-08-27T00:00:00Z" ;
        string variables_name(nvars) ;
                variables_name:long_name = "observation UFO/IODA standard name" ;
        float x_grid(Location) ;
                x_grid:long_name = "observation fractional x-grid location" ;
        float y_grid(Location) ;
                y_grid:long_name = "observation fractional y-grid location" ;
  } // group MetaData
```



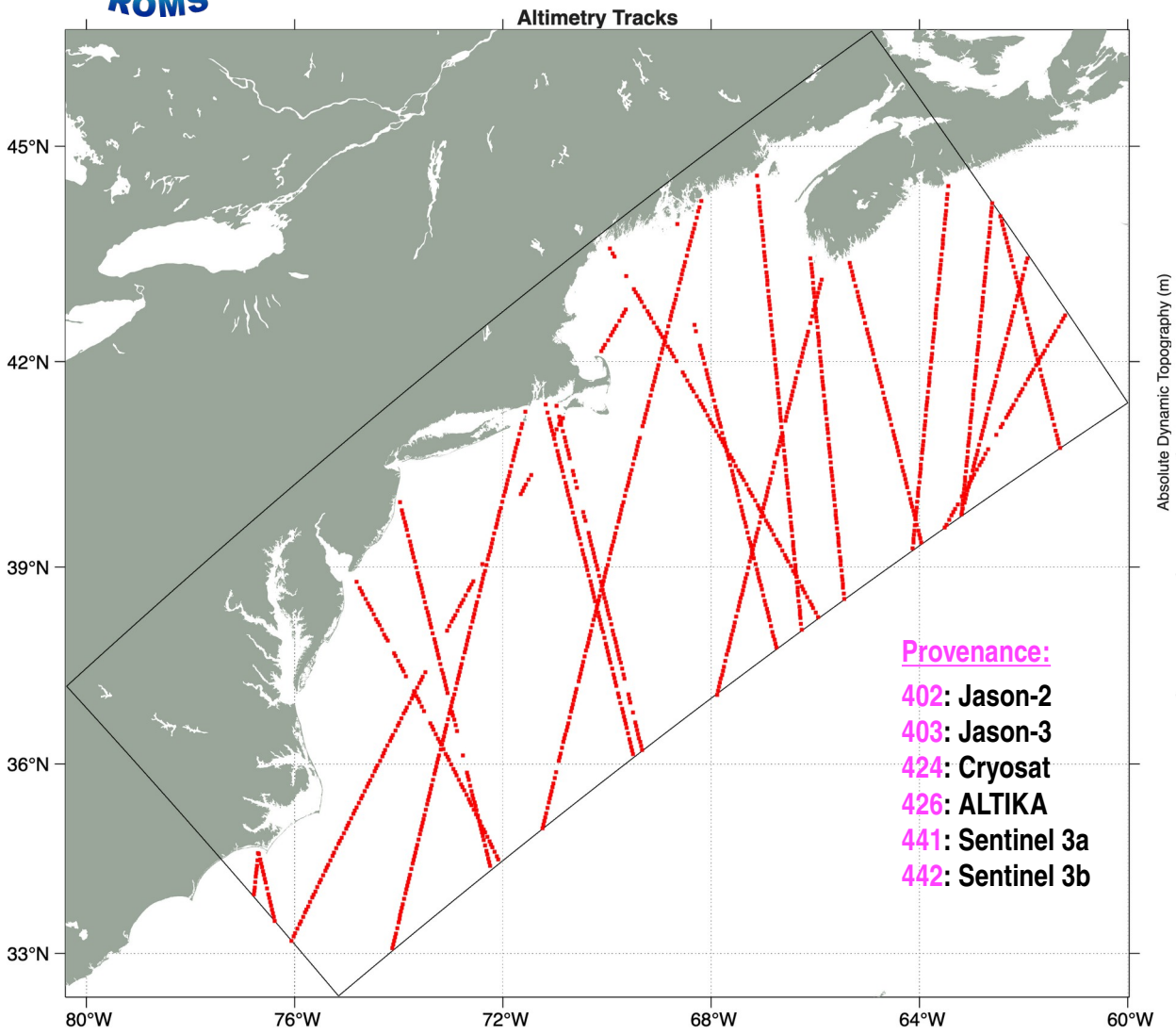
```
group: ObsError {
  variables:
        float absoluteDynamicTopography(Location) ;
                absoluteDynamicTopography:long_name = "observation error standard deviation" ;
                absoluteDynamicTopography:units = "meter" ;
                absoluteDynamicTopography:coordinates = "longitude, latitude, dateTime" ;
  } // group ObsError

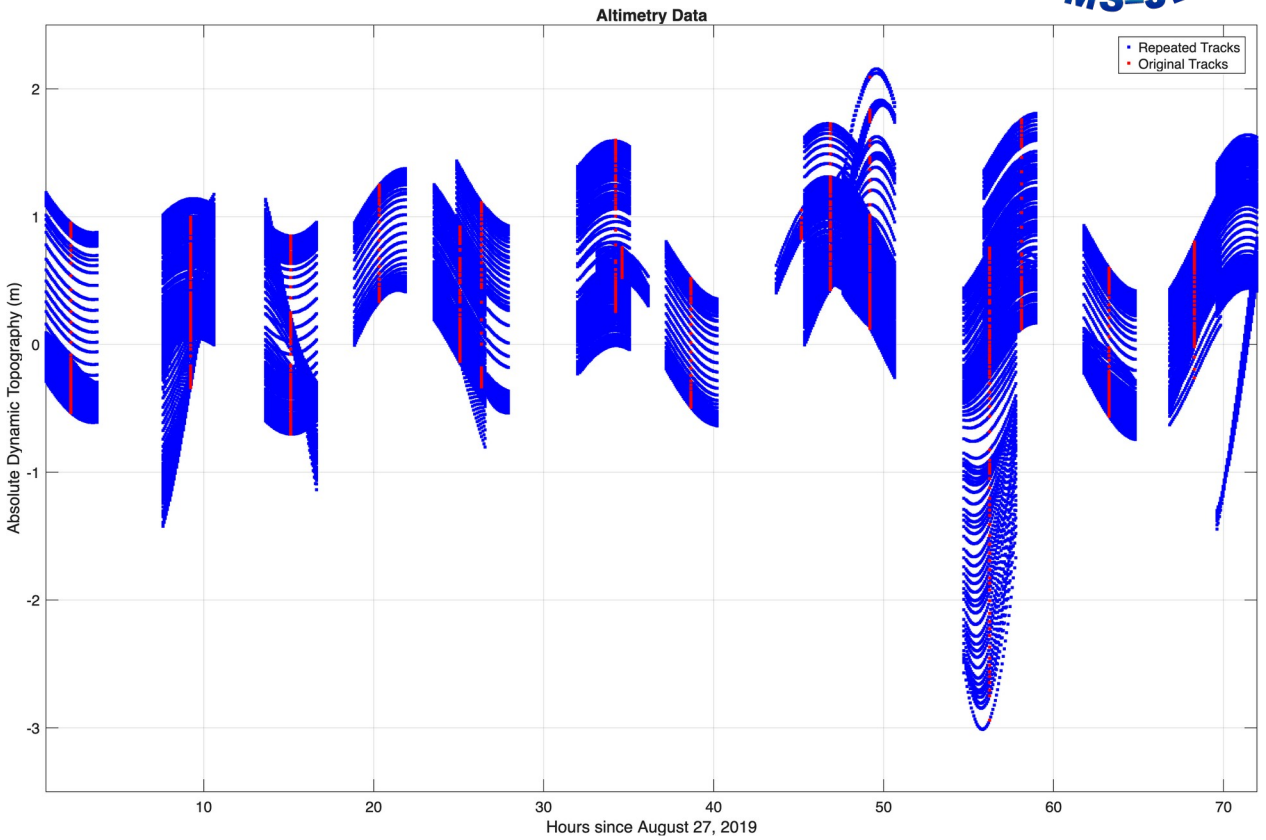
group: ObsValue {
  variables:
        float absoluteDynamicTopography(Location) ;
                absoluteDynamicTopography:long_name = "observation value" ;
                absoluteDynamicTopography:units = "meter" ;
                absoluteDynamicTopography:coordinates = "longitude, latitude, dateTime" ;
  } // group ObsValue

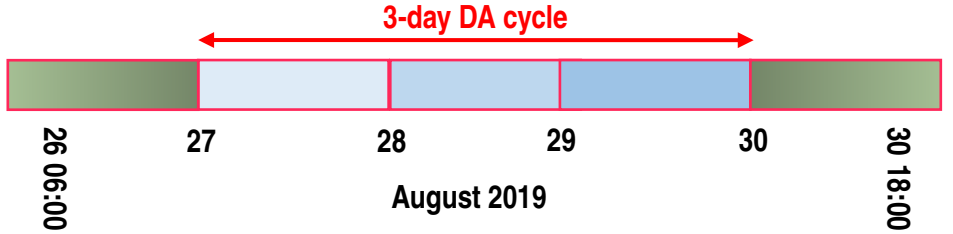
group: PreQC {
  variables:
        int absoluteDynamicTopography(Location) ;
                absoluteDynamicTopography:long_name = "observation preset quality control flag" ;
                absoluteDynamicTopography:coordinates = "longitude, latitude, dateTime" ;
  } // group PreQC
}
```

- The consolidation strategy involves creating multiple IODA-type observation files that can be utilized in native ROMS 4D-Var and ROMS-JEDI data assimilation algorithms for each analysis cycle.
- Time is measured as integers (int64) in seconds from the reference date to avoid rounding errors. We recommend using the starting date of the data assimilation cycle as a reference for small numbers.
- Use single precision (float) for observation data, as they are not measured with high accuracy, resulting in smaller observation files.
- Recommend extension .nc4 for the Enhanced NetCDF-4 observation files.
- It includes optional variables for spatial averaging and/or time averaging filters when computing the nonlinear and linear H(x) operators that calculate the model state vector at the observation locations.
- Use the roms2ioda.m Matlab script to convert from native to multiple IODA observation files, check: github.com/myroms/roms-jedi/tree/develop/tools/obs

Altimetry Data Processing
Absolute Dynamic Topography



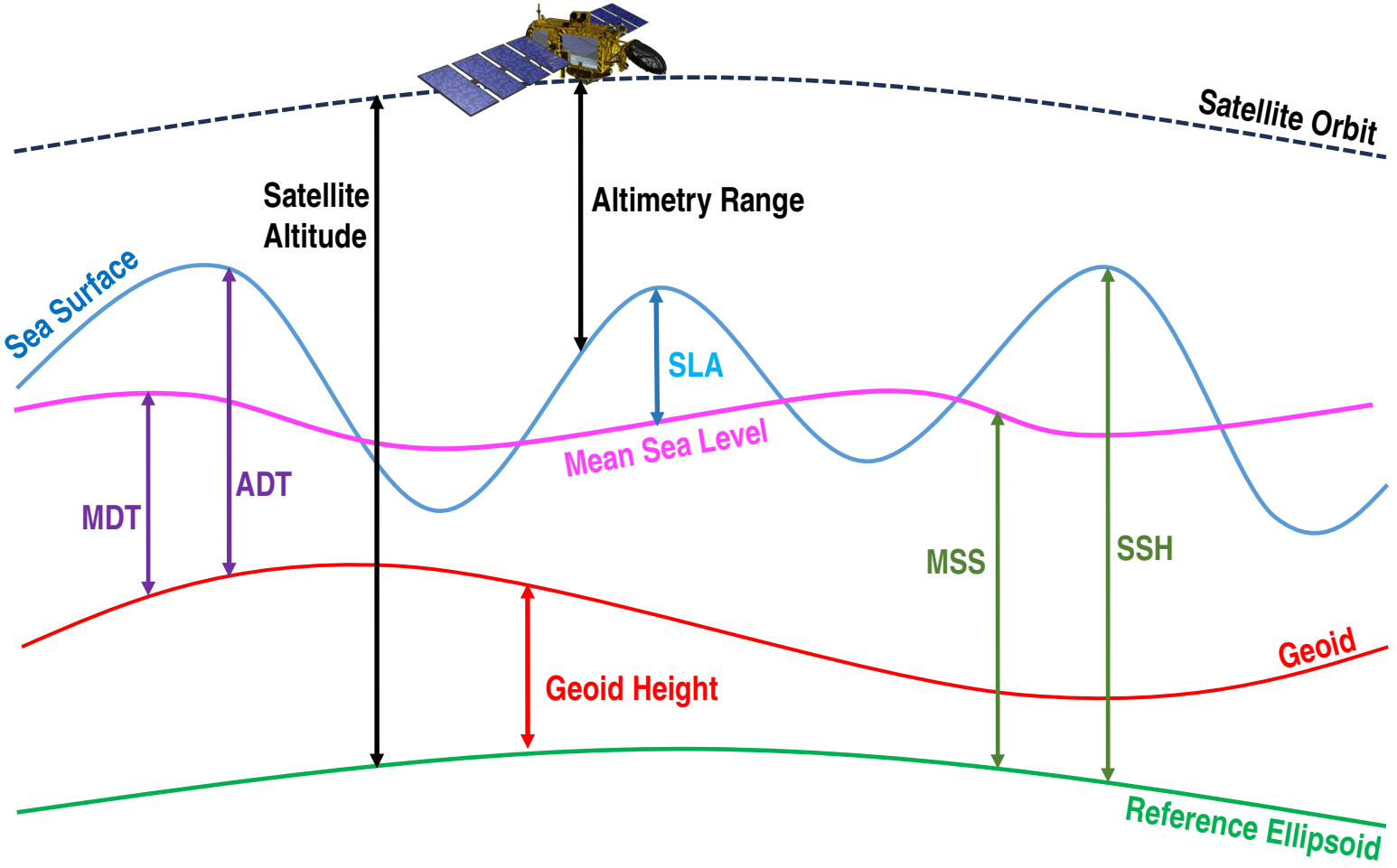




36 hours Time Averaging

Altimetry Observations: Free Surface vs SSH

The ROMS free-surface (ζ, zeta variable) is a time-evolving, physical, dynamic boundary condition at the top of the ocean numerical grid. At the same time, SSH is an instantaneous geodesic measurement by satellite altimeters relative to a reference ellipsoid containing information about the Geoid, ocean dynamics, and seawater steric effects.



SSH: Sea Surface Height
MSS: Mean Sea Surface
SLA: Sea Level Anomaly
ADT: Absolute Dynamic Topography
MDT: Mean Dynamic Topography

JEDI/UFO has an ADT H(x) operator that uses the height of the sea surface above the static Geoid component.

ADT = MDT + SLA
ADT = SSH – MSS + MDT

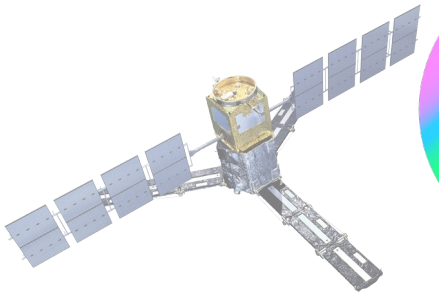
SSH = MSS + SLA
SSH = ADT – MDT+ MSS
SSH = Altitude - Range

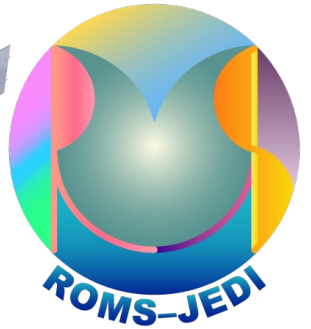


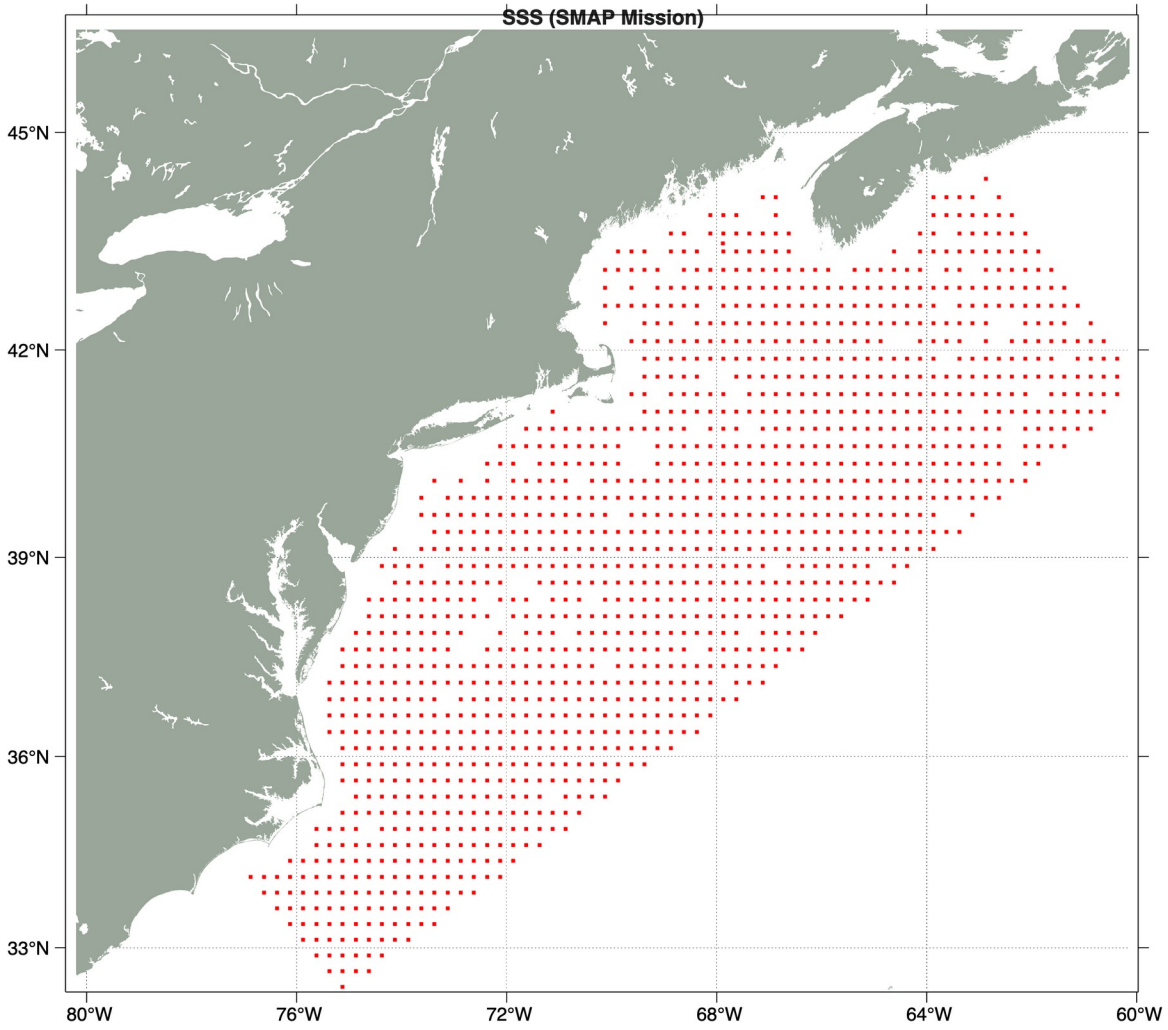


Satellite Sea Surface Salinity
SMAP and SMOS

SMAP: Soil Moisture Active/Passive Satellite (NASA)
SMOS: Soil Moisture and Ocean Salinity Satellite (ESA)





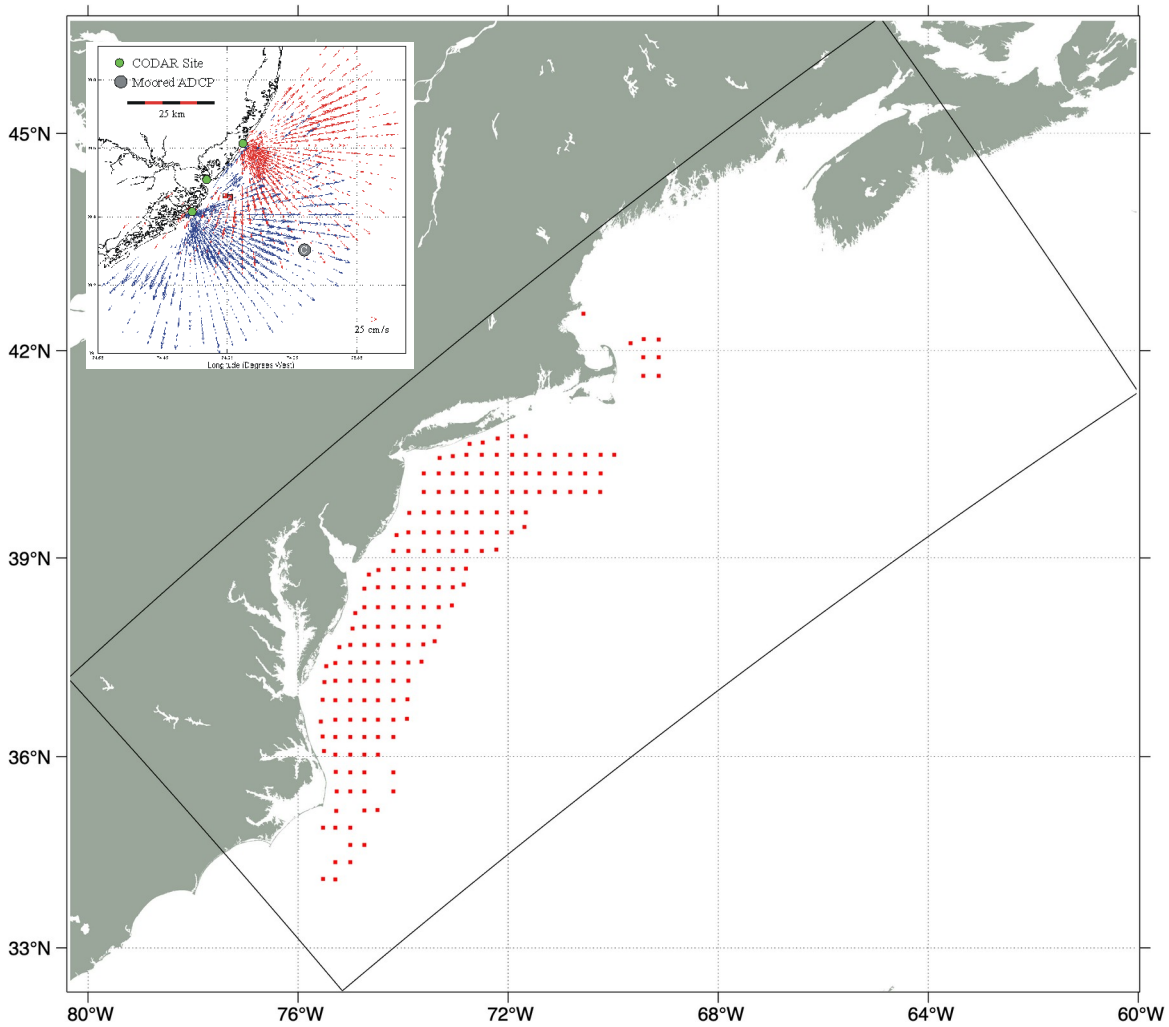


- The SMAP Sea Surface Salinity (SSS) data, Level 2C, is a biased, smooth, and coarse daily composite product.
- The SSS observation operator must include area averaging and time averaging when computing H(x) in ROMS to prevent suppressing the eddy activity observed in the prior near-surface salinity in high-resolution applications.
- Optimally, the data assimilation cost function, *J*, must include a bias term:

)

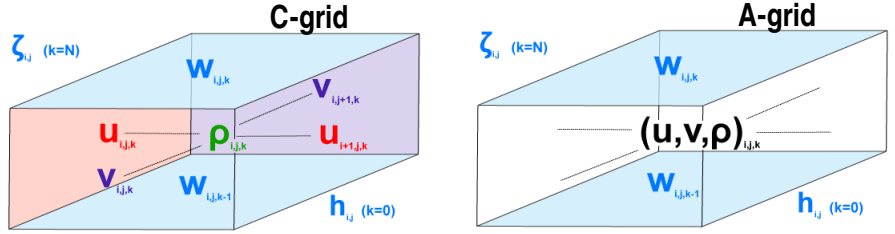
state control vector
background estimate of
background error covariance matrix
observation vector
observation error covariance matrix
observation operator
bias vector
bias field sampled at observation locations
background estimate of
bias error covariance matrix

HF Radar Near Surface Currents
CODAR



- The High-Frequency Radar near-surface currents are obtained from the gridded, hourly product (dataset=hfradar_usegc_6km) distributed by the HFRnet service: https://dods.ndbc.noaa.gov/thredds/catalog/hfradar.html
- The velocity components (m/s) are expected at the grid center (A-grid) and in geographical directions of EAST and NORTH. The concept of staggered velocities is solely used for numerical accuracy and to suppress spurious modes.
- In ROMS, the CODAR velocities are assumed to be located at a constant depth, Zr = -2 m.
- We commend using an area-averaged and time-averaged operator in the computation of H(x) at the observation locations to suppress inertial oscillations and tides.
- A nonlinear and linear variables change operator is required to convert between the A-grid and the C-grid, and vice versa.

Ocean Currents Variable Changes



C-grid to A-grid Currents

- **Average to RHO points:**

  Urho(i,j,k) = 0.5 * (Uc(i,j,k) + Uc(i+1,j,k)) i = Istr: Iend

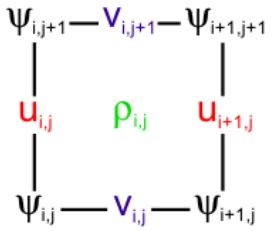
  Vrho(i,j,k) = 0.5 * (Vc(i,j,k) + Vc(i,j+1,k)) j = Jstr:Jend

- **Apply gradient lateral boundary conditions on the normal boundary and rotate to Eastward and Northward directions:**

  Ua(i,j,k) = Urho(i,j,k) * CosAngler(i,j) - Vrho(i,j,k) * SinAngler(i,j)

  Va(i,j,k) = Vrho(i,j,k) * CosAngler(i,j) + Urho(i,j,k) * SinAngler(i,j)

C-grid



A-grid to C-grid Currents

- **Rotate to computational ξ and η grid directions:**

  Urot(i,j,k) = Ua(i,j,k) * CosAngler(i,j) + Va(i,j,k) * SinAngler(i,j)
  Vrot(i,j,k) = Va(i,j,k) * CosAngler(i,j) - Ua(i,j,k) * SinAngler(i,j)

- **Average to staggered C-grid locations:**

  Uc(i,j,k) = 0.5 * (Urot(i-1,j,k) + Urot(i,j,k)) i = Istr: IendR
  Vc(i,j,k) = 0.5 * (Vrot(i,j-1,k) + Vrot(i,j,k)) j =Jstr:JendR

