Introducing the ROMS-JEDI Interface

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

[1]Rutgers University, DMCS, New Brunswick, NJ, USA
[2]Univesity of California, Santa Cruz, CA,USA
arango@marine.rutgers.edu

Consiglio Nazionale delle Ricerche

Ocean Predict DA-TT Meeting,

Rome, Italy

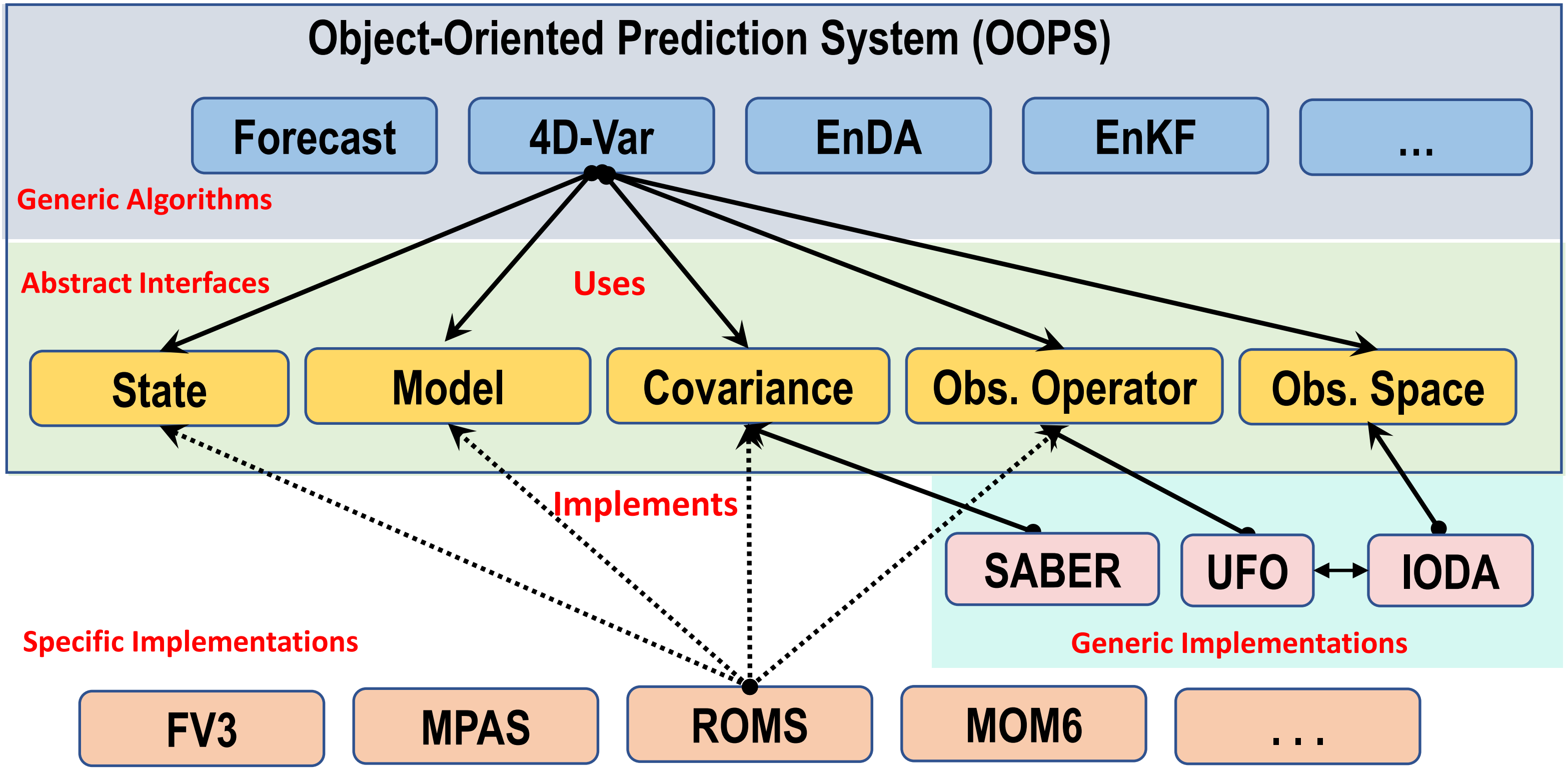
May10, 2023

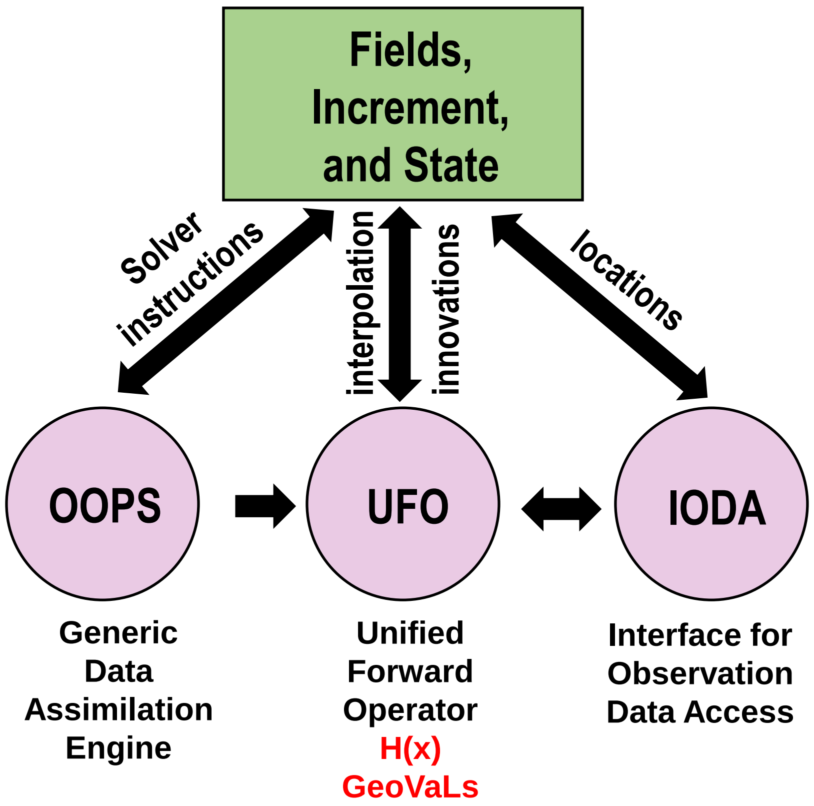
ROMS-JEDI Interface

BUMP	B matrix on an Unstructured Mesh Package
EWOK	Experiments and Workflows Orchestration Kit
FGAT	First Guess at Appropriate Time
IODA	Interface for Observation Data Access
JCSDA	Joint Center for Satellite Data Assimilation
JEDI	Joint Effort for Data Assimilation Integration
LETKF	Local Ensemble Transform Kalman Filter
NICAS	Normalized Interpolated Convolution from an Adaptive Subgrid
OOPS	Object-Oriented Prediction System
R2D2	Research Repository for Data and Diagnostics
SABER	System-Agnostic Background Error Representation
UFO	Unified Forward Operator
VADER	The VAriable DErivation Repository
3D-Var	3-Dimensional Variational Data Assimilation
4D-Var	4-Dimensional variational Data Assimilation

ROMS-JEDI Integration

JEDI Motto and Catchphrase: Separation of Concerns is the most crucial part of its design and implementation!





Modern, abstract object-oriented programming data assimilation framework written in C++, because of its robust template capabilities. OOPS was designed initially at ECMWF, now expanded and maintained by JCSDA.

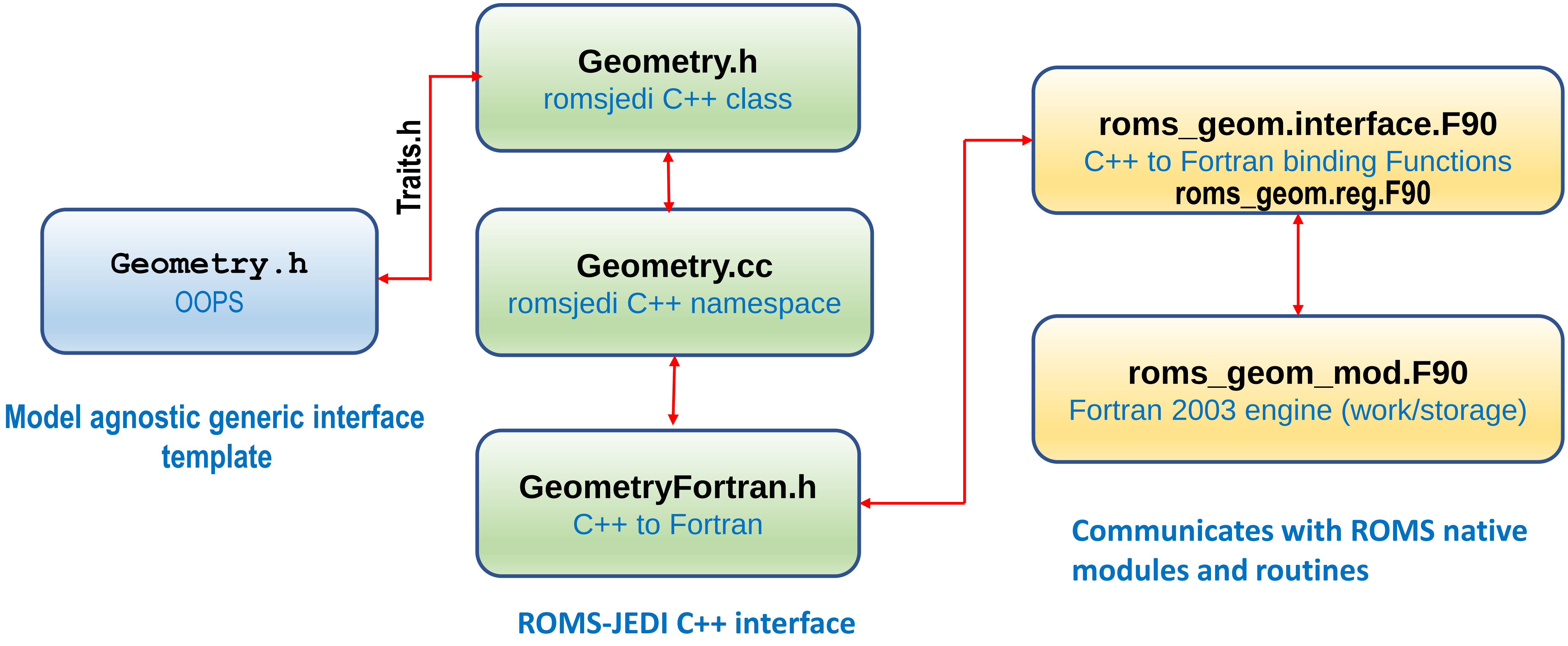
ROMS-JEDI Interface Classes

Mostly any modern forecast model can be connected to the JEDI framework, provided that the following predefined, abstract C++/Fortran Classes or building blocks are coded:

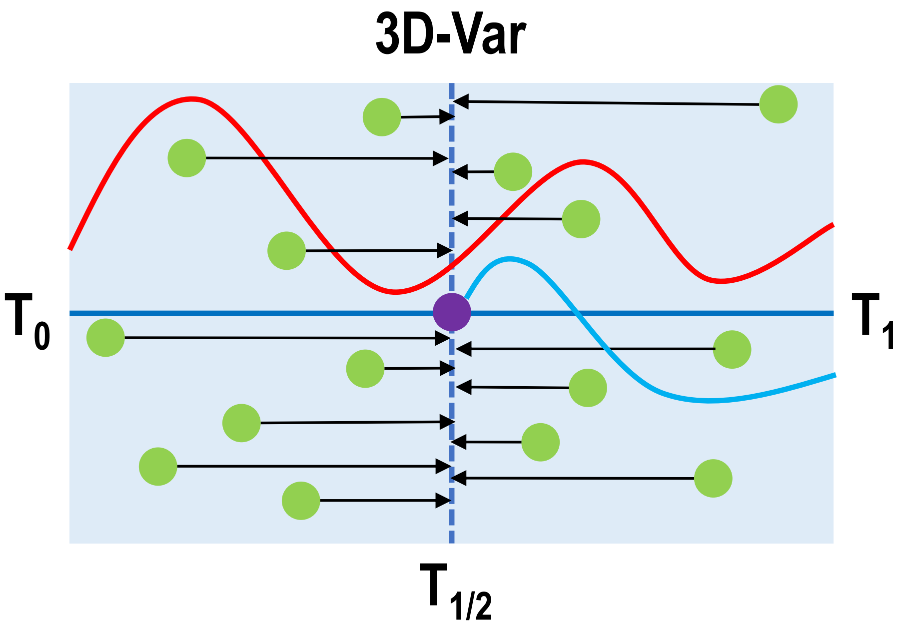
CLASS	DESCRIPTION
ErrorCovariance	Background error covariance and modeling (**SABER**: BUMP/NICAS)
Field/Fields	Elemental operators to manipulate a field or a set of fields to the model state/increment vector and metadata
Geometry	Application grid definition, including coordinates, metrics, parallel decomposition, and operators
GeometryIterator	Methods to set/get state fields over specified grid points in **LETKF** applications
Increment	Procedures to operate on the increment vector that extends/inherits from the **Fields** class
LinearModel	Initializes, run, and finalizes the Tangent Linear and Adjoint model dynamical/numerical kernels
LinearVariableChange	Tangent/adjoint increment vector variables transformation from one field to another
Localization	Model Ensemble Localization (**SABER**: BUMP/NICAS)
Model	Initializes, run, and finalizes the Nonlinear model dynamical/numerical kernel
State	Procedures to operate on the state vector that extends/inherits from the **Fields** class
Trajectory	Methods to process the nonlinear trajectory that linearizes the tangent linear and adjoint models
VariableChange	Nonlinear state vector variables transformation from one field to another

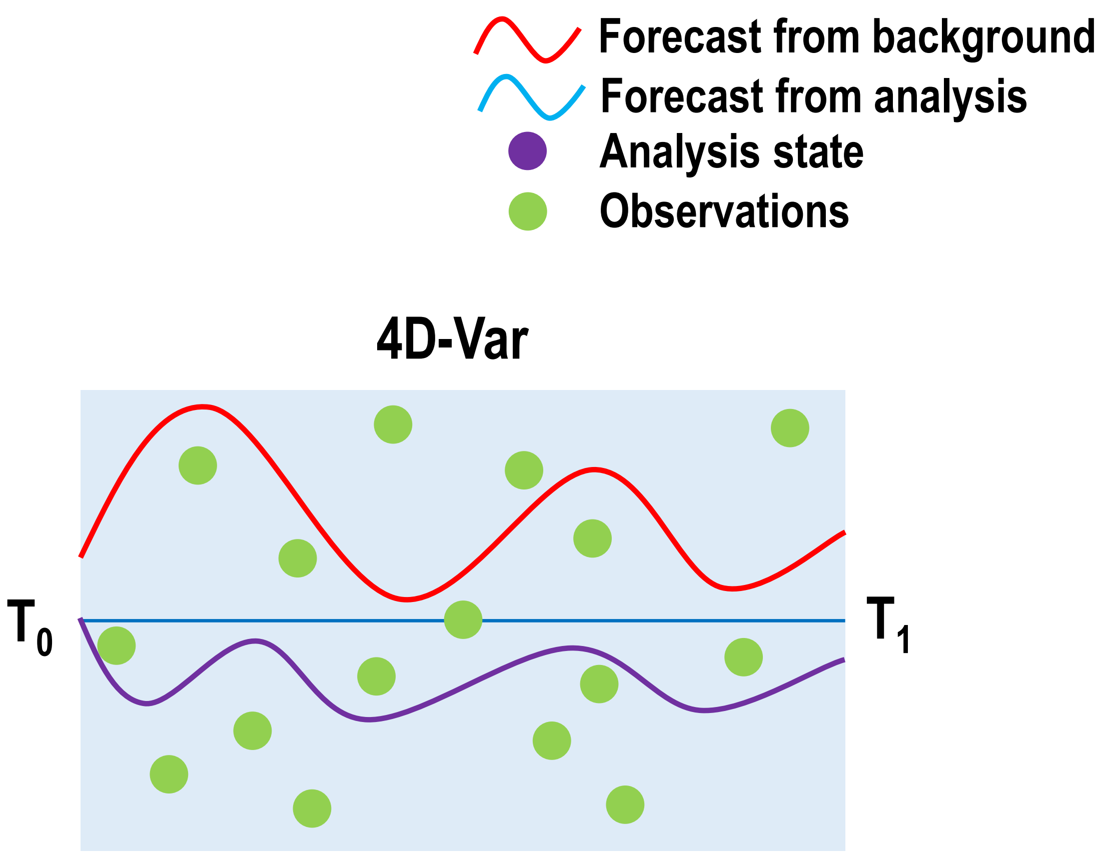
C++ to Fortran to C++ Binding Sequence

Interoperability mechanism for the Geometry Class that allows Fortran to invoke C++ functions and vice versa C++ to invoke Fortran procedures



Variational Data Assimilation Cost Functions:

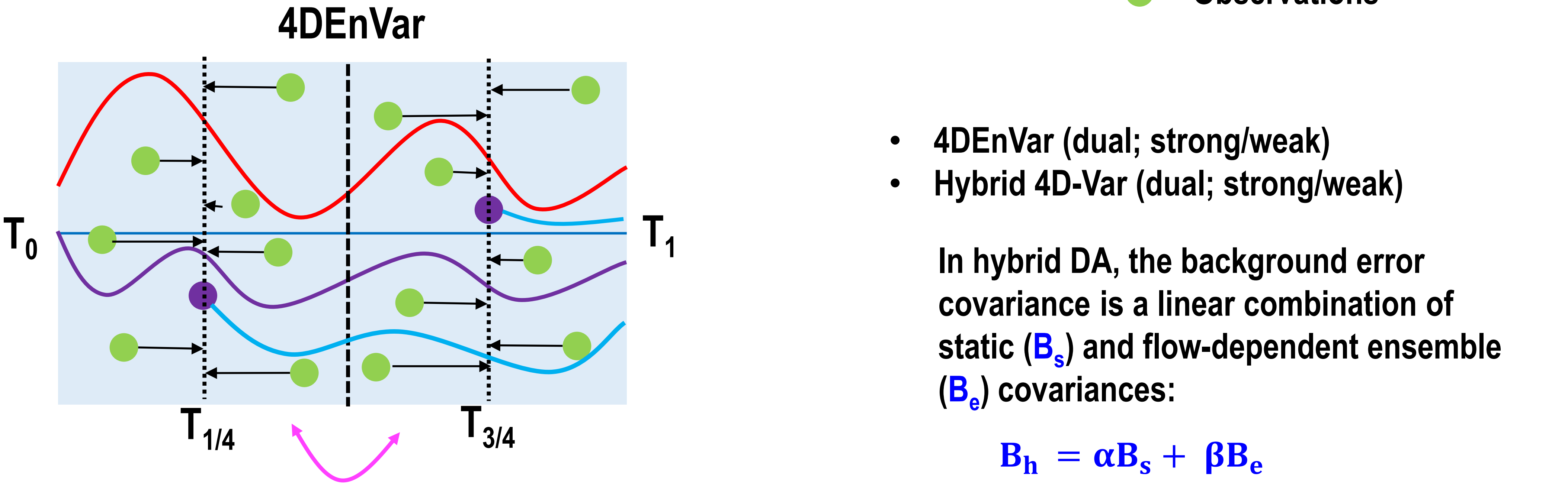




- 3D-Var (primal and dual)
- 3DEnVar (primal and dual)
- Hybrid 3D-Var (primal and dual)
- Identity Tangent Linear and Adjoint Operators

- 3D-Var FGAT (primal and dual)
- 3DEnVar FGAT (primal and dual)
- Hybrid 3D-Var FGAT (primal and dual)
- 4D-Var (primal and dual; strong/weak)

Variational Data Assimilation Cost Functions:



B matrix cross terms

- 4DEnVar (dual; strong/weak)
- Hybrid 4D-Var (dual; strong/weak)

In hybrid DA, the background error covariance is a linear combination of static ($\mathbf{B}_s$) and flow-dependent ensemble ($\mathbf{B}_e$) covariances:

$$\mathbf{B}_h = \alpha \mathbf{B}_s + \beta \mathbf{B}_e$$

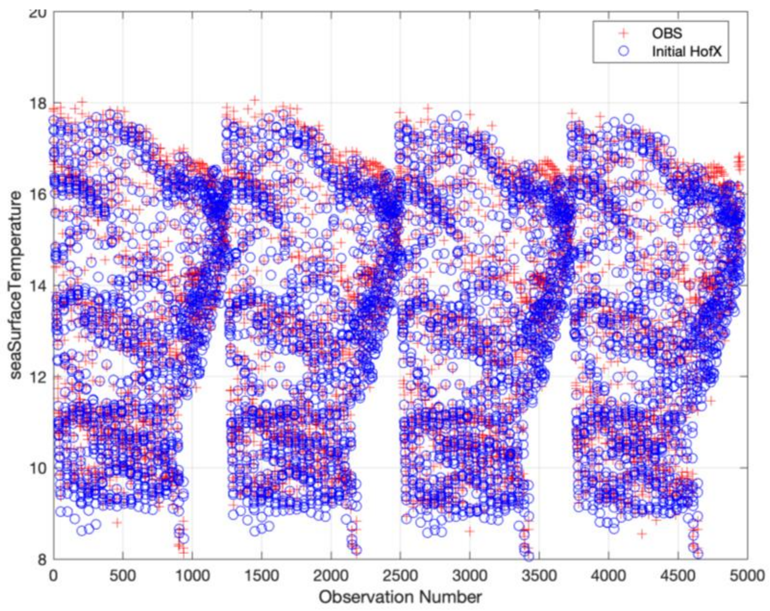
The α and β coefficients are appropriately chosen to build the hybrid background error covariance

OOPS Variational Data Assimilation Minimizers:

	Minimizer	Description
	DRGMRESR	Derber-Rosati GMRESR solver
	DRIPCG	Derber-Rosati Inexact-Preconditioned Conjugate Gradient solver
	DRPCG	Derber-Rosati Preconditioned Conjugate Gradient solver
	DRPFOM	Derber-Rosati Preconditioned Full Orthogonal Method (FOM) solver
Primal	**DRPLanczos**	Derber-Rosati Preconditioned Lanczos solver
	FGMRES	Flexible GMRES solver, Saad and Schultz (1986), Saad (1993)
	GMRESR	GMRESR solver, Van der Vorst and Vuik (1994)
	IPCG	Inexact-Preconditioned Conjugate Gradient, Golub and Ye (1999, 2000)
	LBGMRESR	Left B Preconditioned GMRESR solver
	MINRES	MINRES solver, Paige and Saunders (1975)
	PCG	Preconditioned Conjugate Gradient solver
	PLanczos	Preconditioned Lanczos solver
Dual	**RPCG**	Restricted Preconditioned Conjugate Gradient, Gratton and Tshimanga (2009)
	RPLanczos	Augmented Restricted Lanczos method, Gürol (2013)
	SaddlePoint	Time-parallelized Saddle-Point formulation, Fisher and Gürol (2017)

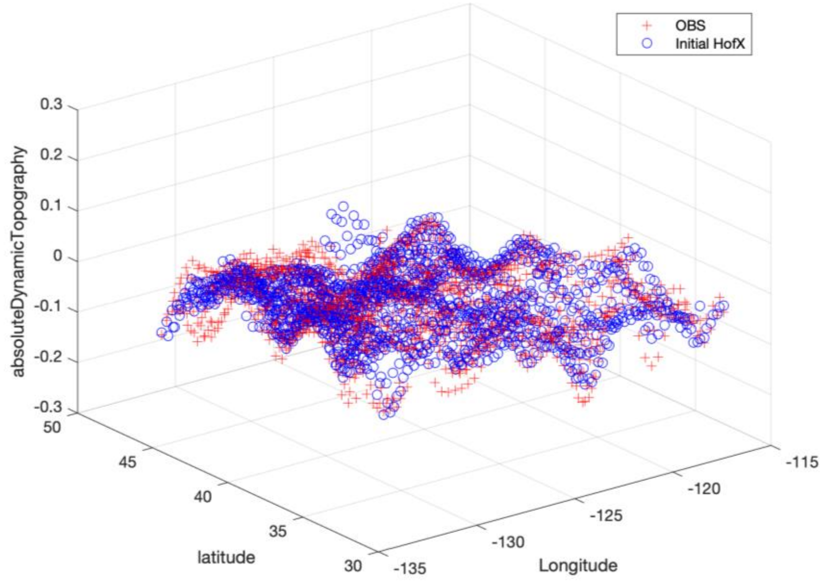
UFO Observation Interpolation, H(X): WC13 Application

Satellite SST (C)

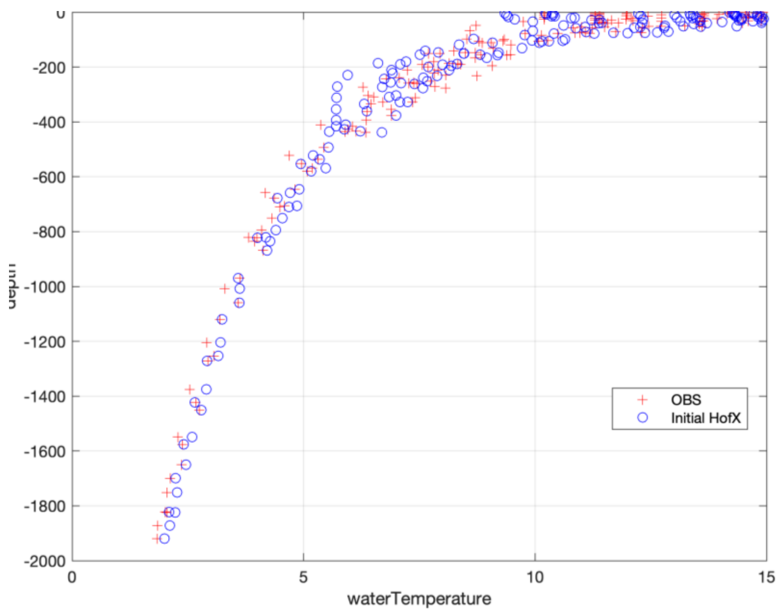


Jan 3 12:00 Jan 4 12:00 Jan 5 12:00 Jan 6 12:00

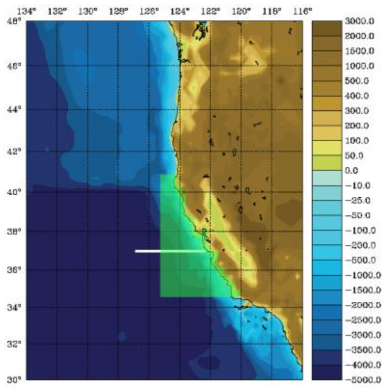
Absolute Dynamic Topography (m)



In situ Temperature (C)



Model Bathymetry with 37°N Transect and Target Area



Assimilating SST, Altimetry, and insitu Temperature and Salinity profiles.

Grid: 54x53x30

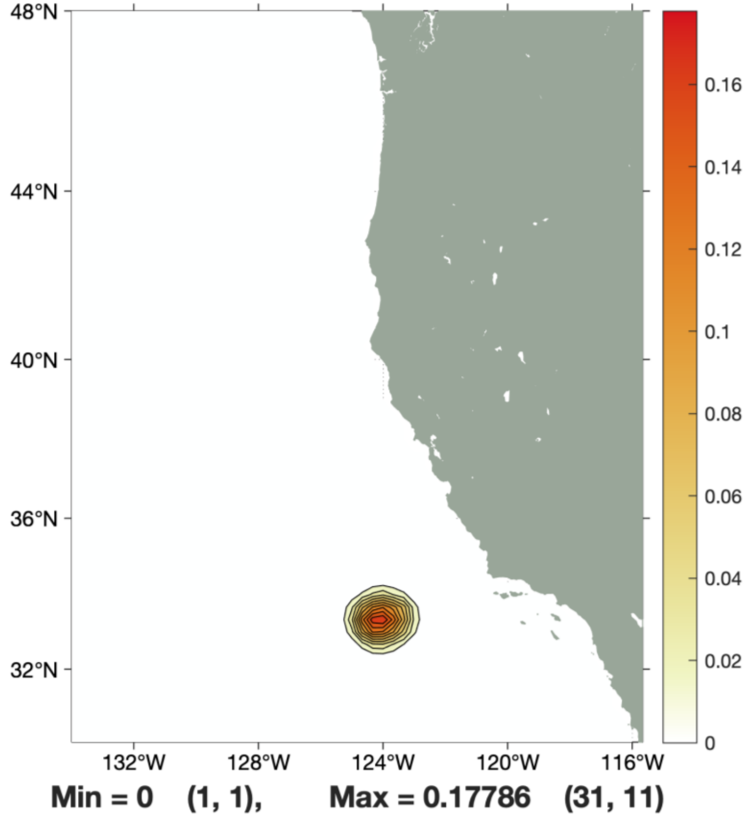
Resolution: 32km x 35km

Cycle: Jan 3-7, 2004

SABER (BUMP/NICAS) Error Covariance B Modeling

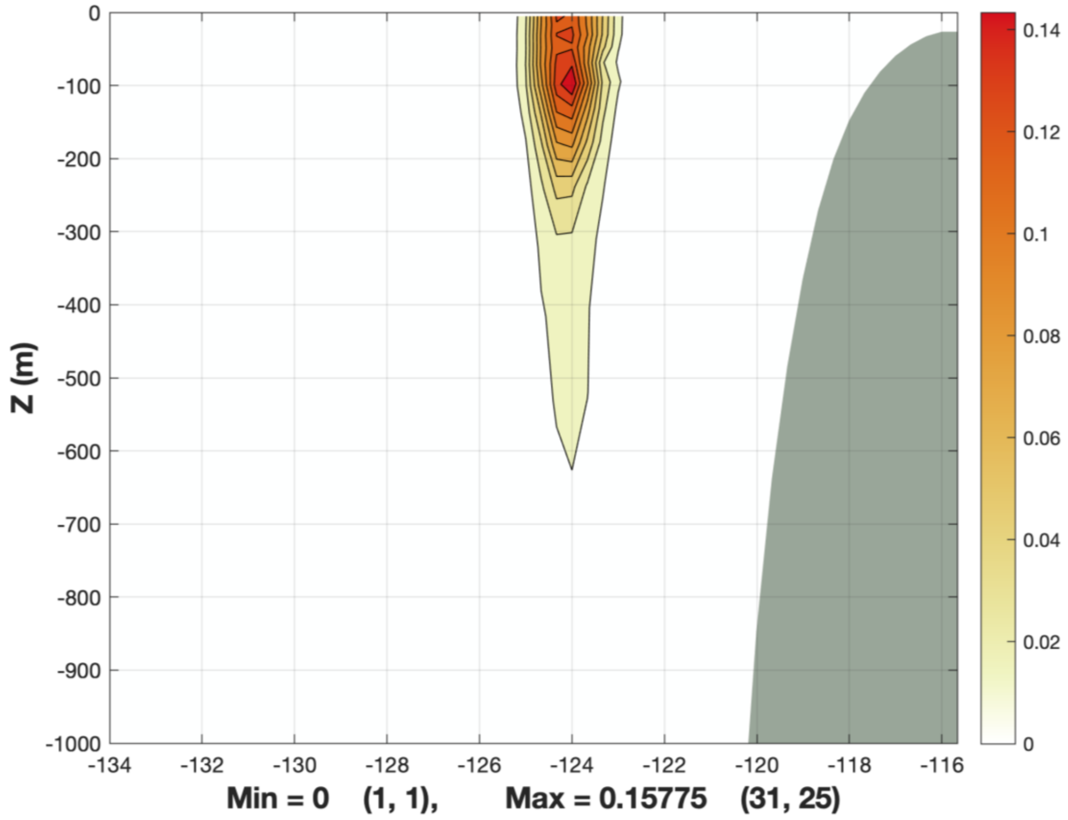
Single observation Dirac Impulse scaled by the standard deviation

Surface Temperature



160 km Horizontal Correlation

Temperature

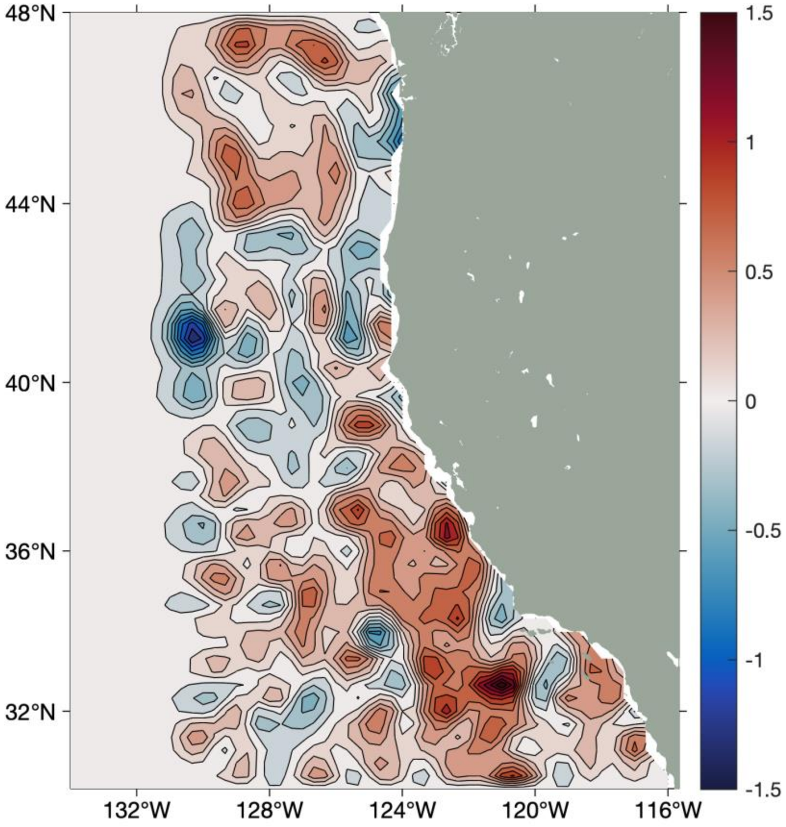


150 m Vertical Correlation

WC13 Data Assimilation Increment:
Surface Temperature

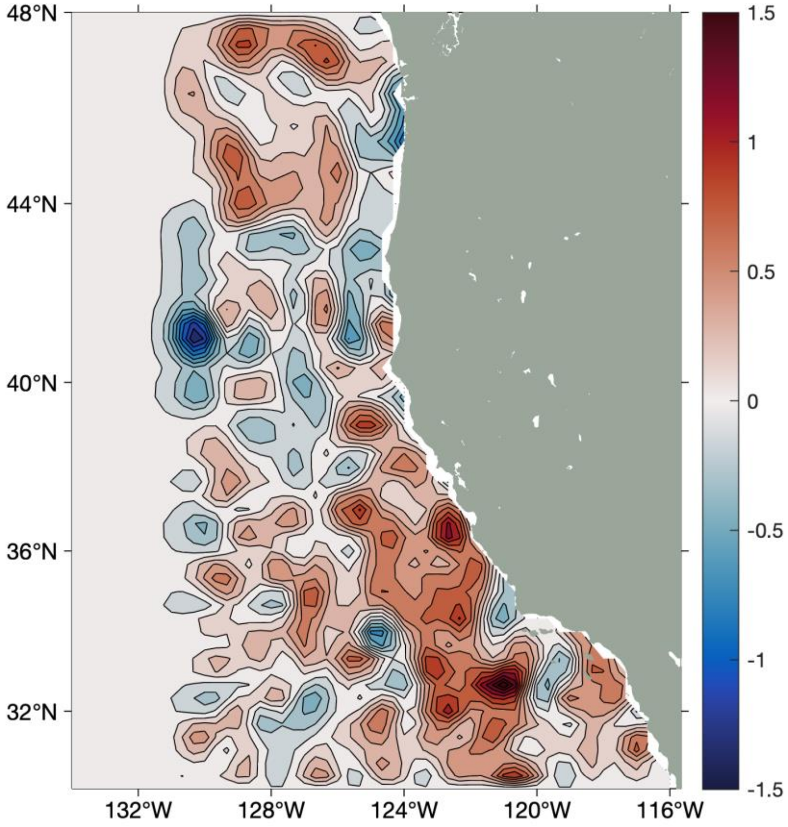
ROMS-JEDI

3D-Var (Primal)



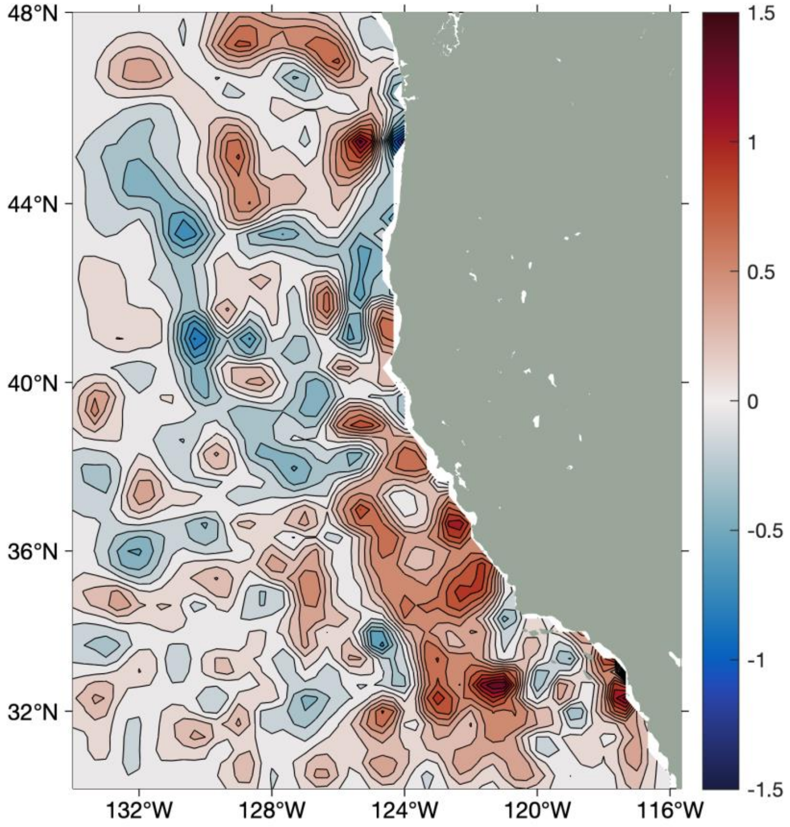
Min = -1.3255 (12, 34), Max = 1.7117 (40, 9)

3D-Var (Dual)



Min = -1.3499 (12, 34), Max = 1.7638 (40, 9)

Native RBL4D-Var (Dual)



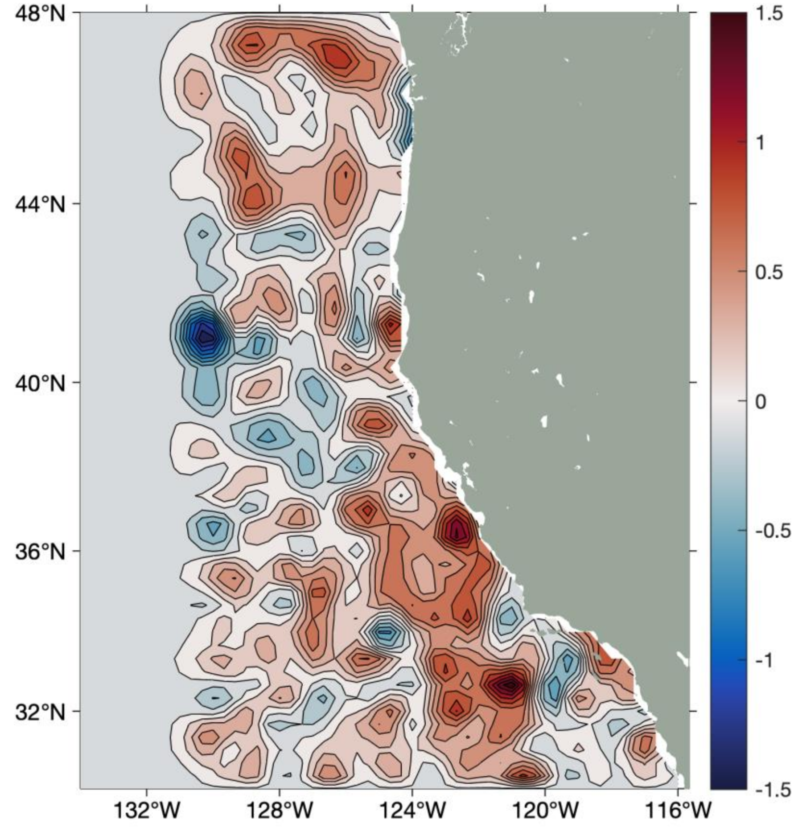
Min = -1.3717 (31, 47), Max = 1.4367 (40, 9)

ROMS TLM and ADM are Identity Operators

Uses ROMS TLM and ADM kernels

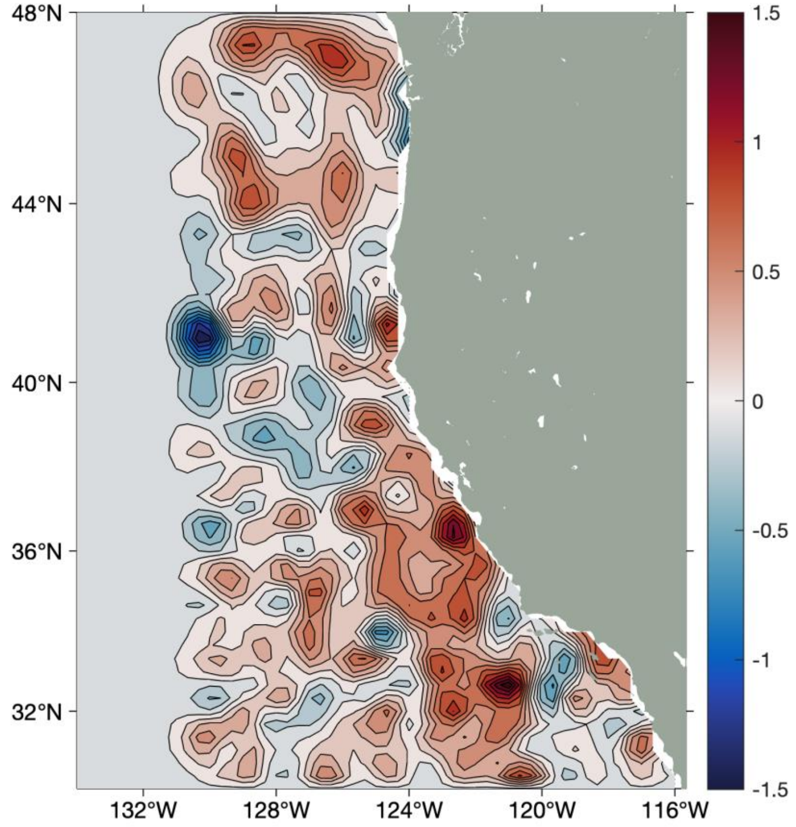
WC13 Data Assimilation Increment:
Surface Temperature

3D-Var FGAT (Primal)



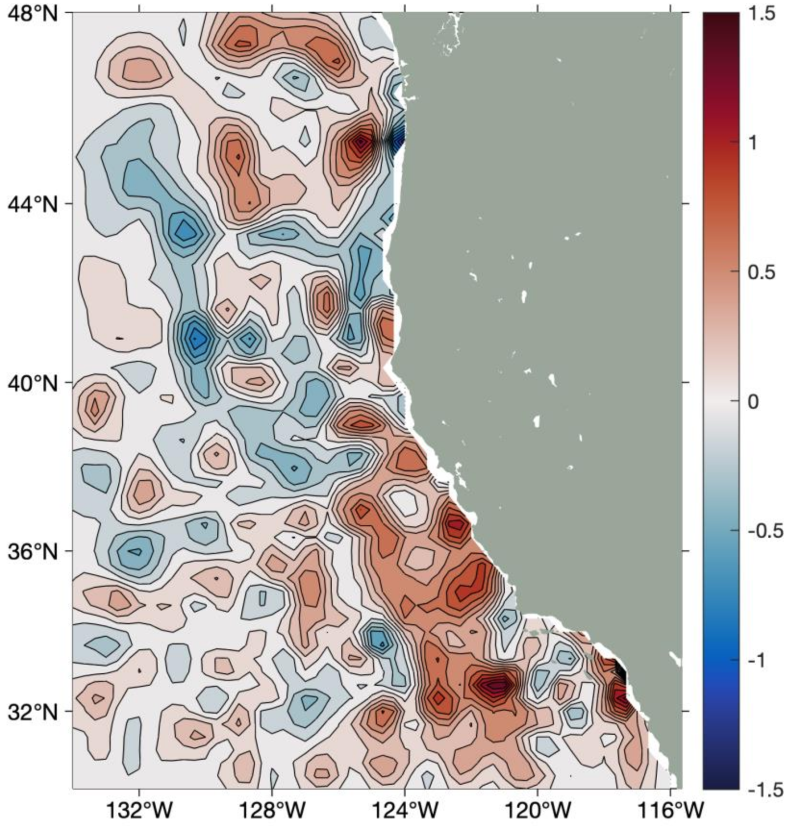
Min = -1.4314 (12, 34), Max = 1.6313 (40, 9)

3D-Var FGAT (Dual)



Min = -1.4446 (12, 34), Max = 1.6722 (40, 9)

Native RBL4D-Var (Dual)



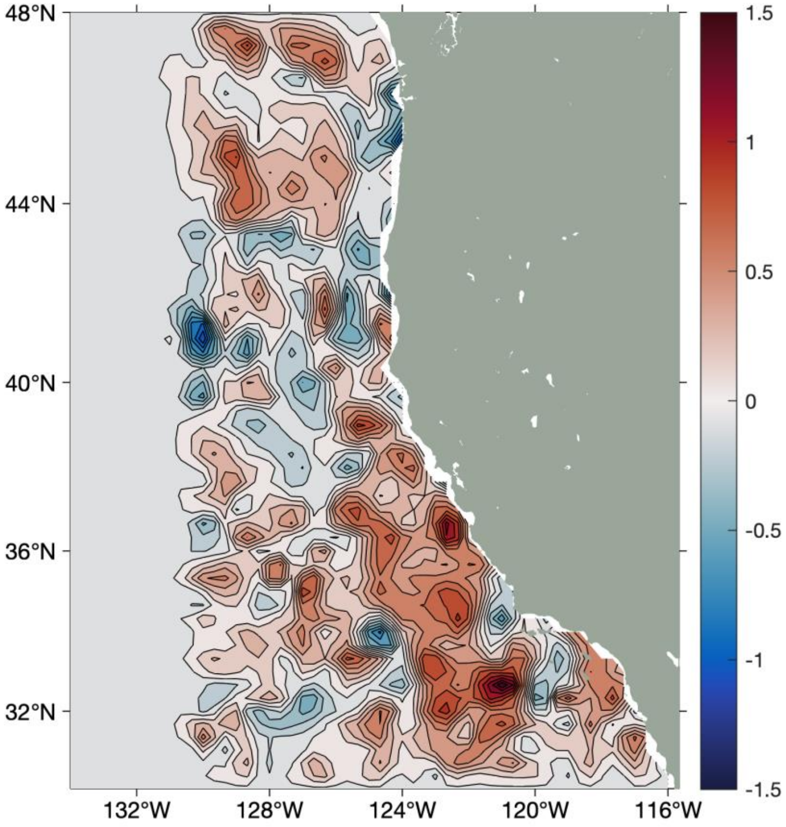
Min = -1.3717 (31, 47), Max = 1.4367 (40, 9)

ROMS TLM and ADM are Identity Operators

Uses ROMS TLM and ADM kernels

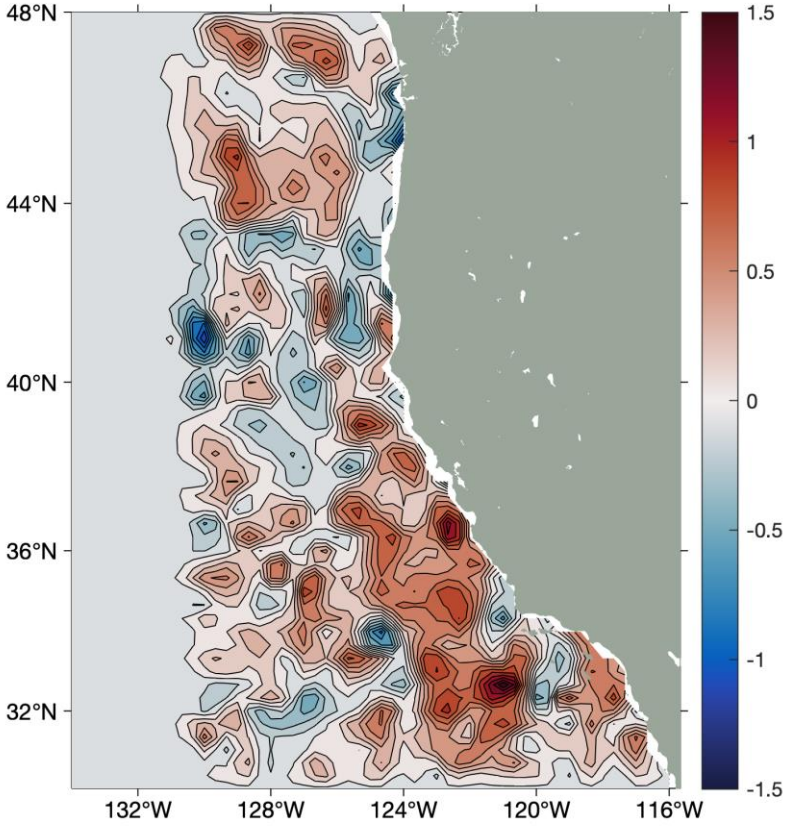
WC13 Data Assimilation Increment:
Surface Temperature

3DEnVar (Primal)



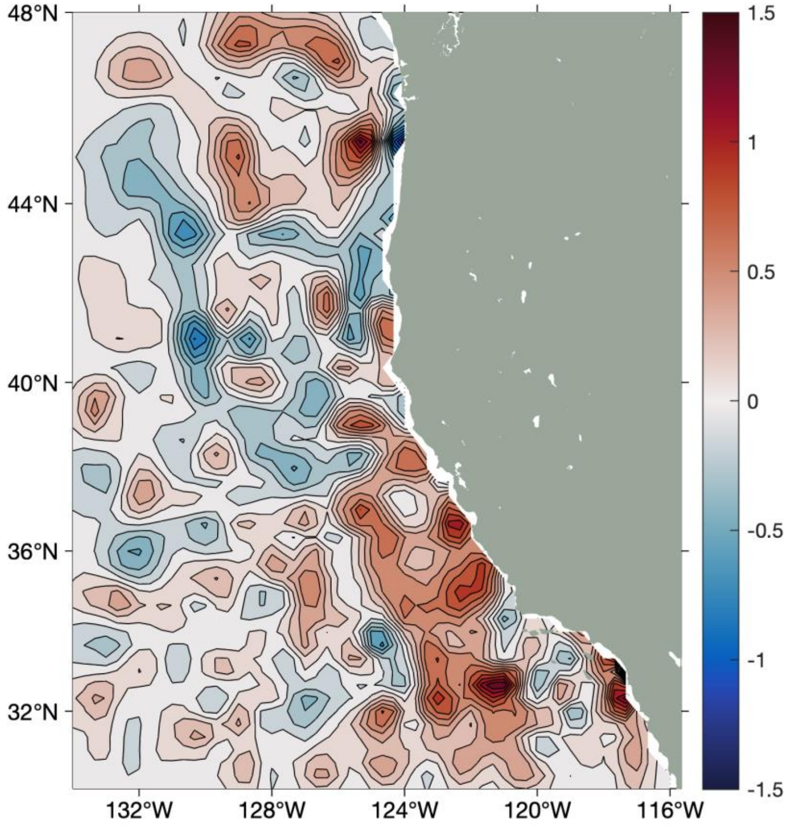
Min = -1.1178 (13, 34), Max = 1.5743 (40, 9)

3DEnVar (Dual)



Min = -1.1493 (13, 34), Max = 1.618 (40, 9)

Native RBL4D-Var (Dual)



Min = -1.3717 (31, 47), Max = 1.4367 (40, 9)

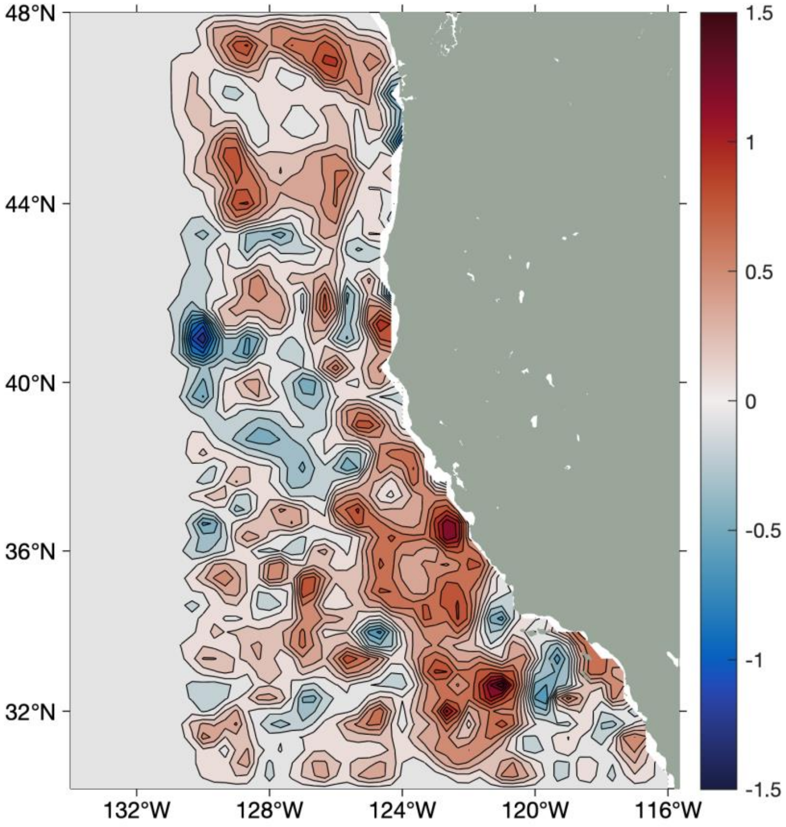
ROMS TLM and ADM are Identity Operators

Ensemble Size = 10

Uses ROMS TLM and ADM kernels

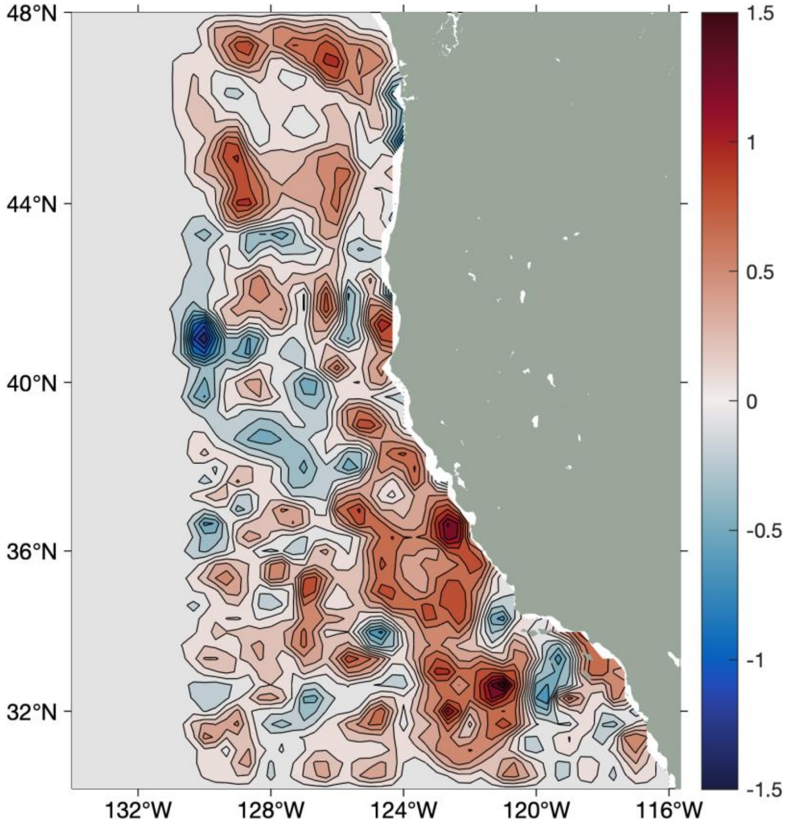
WC13 Data Assimilation Increment:
Surface Temperature

3DEnVar FGAT (Primal)



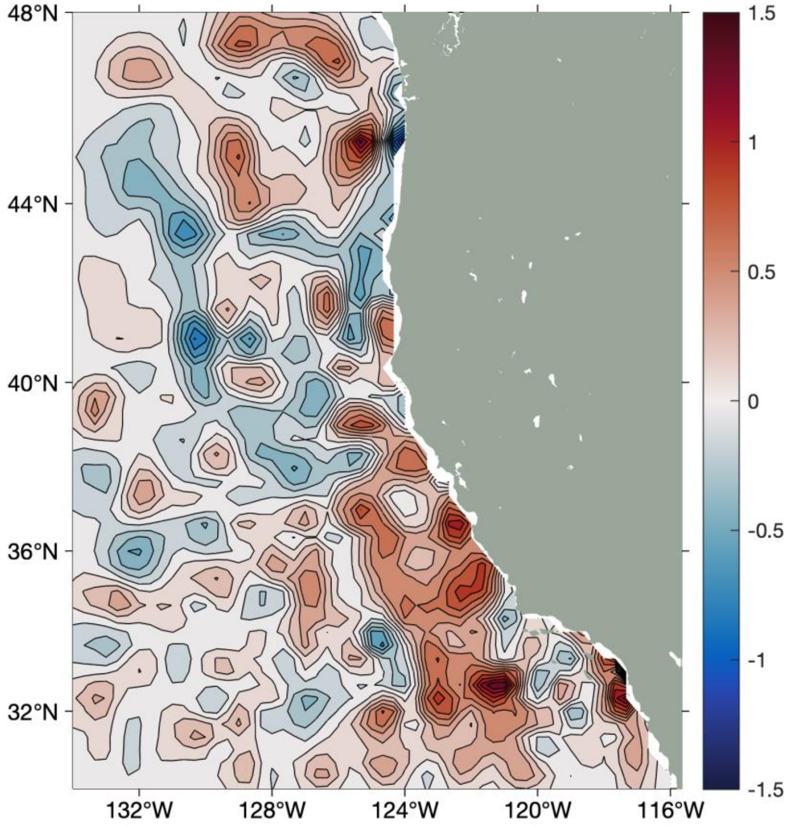
Min = -1.298 (13, 34), Max = 1.5914 (40, 9)

3DEnVar FGAT (Dual)



Min = -1.348 (13, 34), Max = 1.6471 (40, 9)

Native RBL4D-Var (Dual)



Min = -1.3717 (31, 47), Max = 1.4367 (40, 9)

ROMS TLM and ADM are Identity Operators

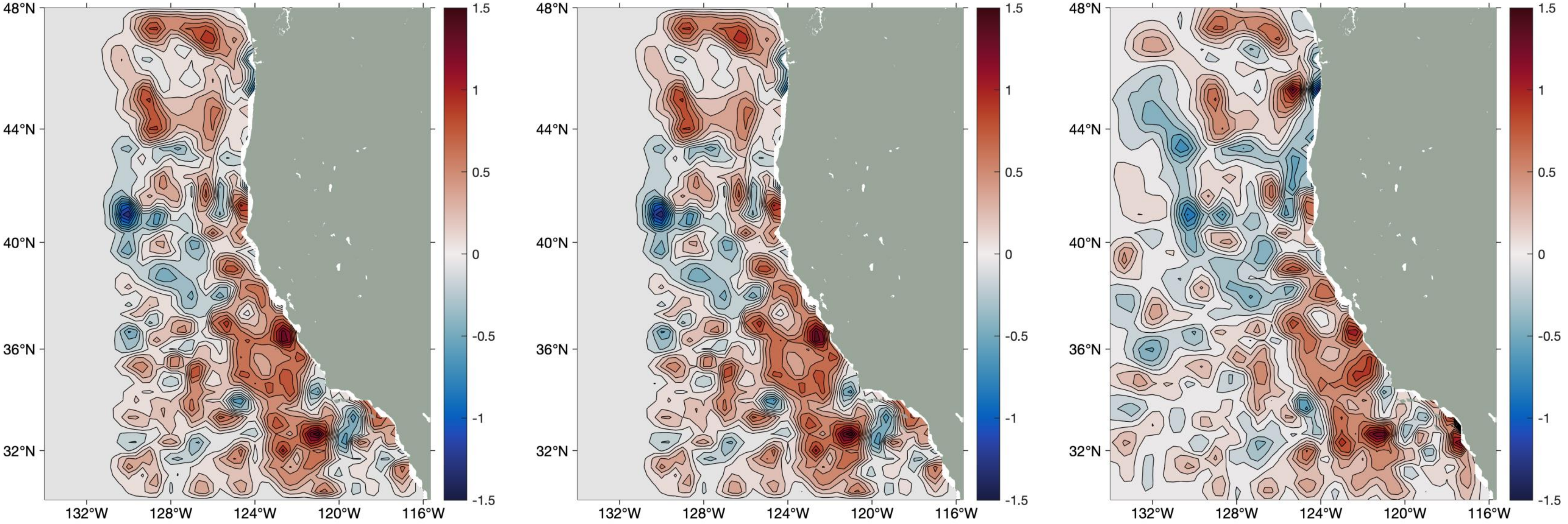
Ensemble Size = 10

Uses ROMS TLM and ADM kernels

WC13 Data Assimilation Increment: Surface Temperature

$\alpha = 0.4$
$\beta = 0.6$

Hybrid 3DEnVar FGAT (Primal) | **Hybrid 3DEnVar FGAT (Dual)** | **Native RBL4D-Var (Dual)**



Min = -1.2965 (13, 34), Max = 1.5823 (40, 9)

Min = -1.3284 (13, 34), Max = 1.6377 (40, 9)

Min = -1.3717 (31, 47), Max = 1.4367 (40, 9)

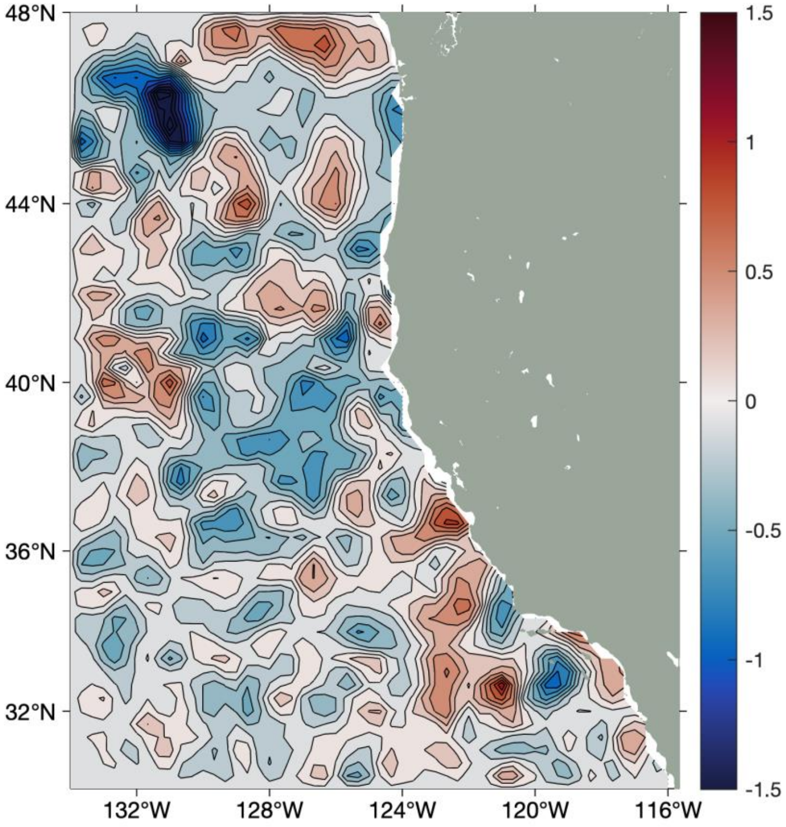
ROMS TLM and ADM are Identity Operators

Ensemble Size = 10

Uses ROMS TLM and ADM kernels

WC13 Data Assimilation Increment:
Surface Temperature

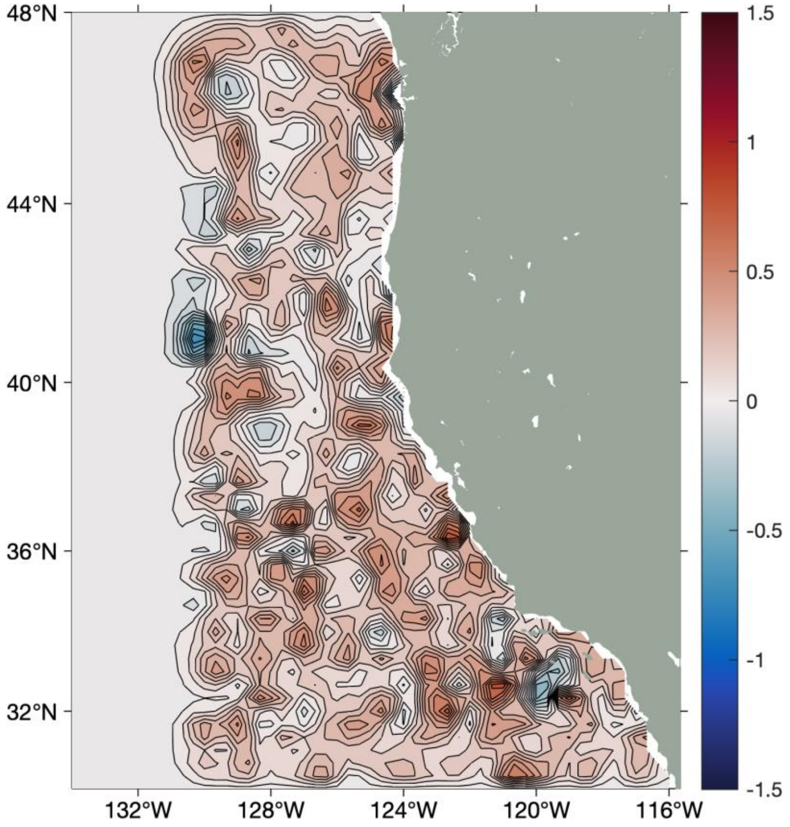
LETKF



Min = -1.8148 (10, 48), Max = 1.2043 (40, 9)

No Inflation
Ensemble Size = 10

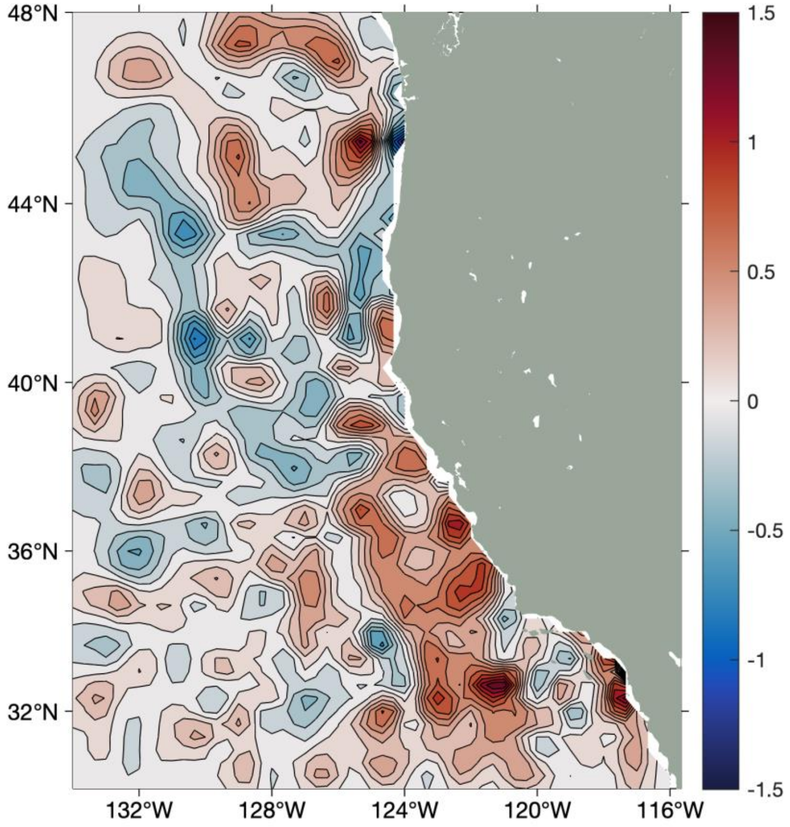
LETKF / Hybrid 3DEnVar FGAT
Penny (2014; MWR)



lin = -0.69029 (12, 34), Max = 0.83542 (35, 20)

Identity TLM/ADM
Ensemble Size = 10

Native RBL4D-Var (Dual)

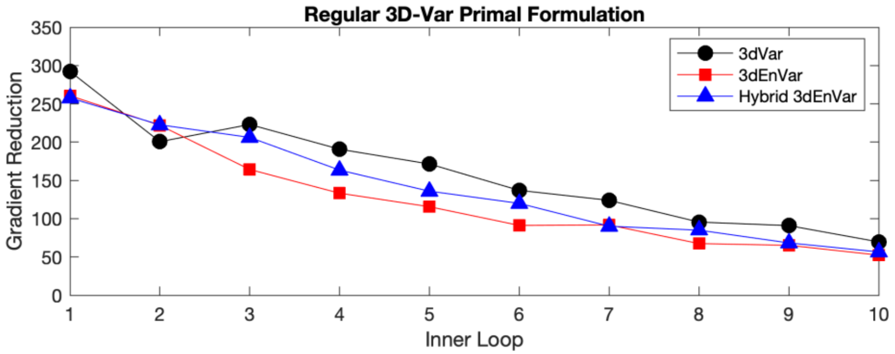


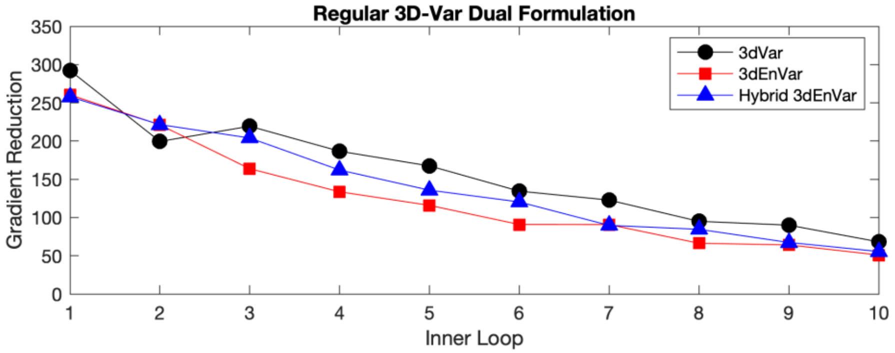
Min = -1.3717 (31, 47), Max = 1.4367 (40, 9)

Uses ROMS TLM and ADM kernels

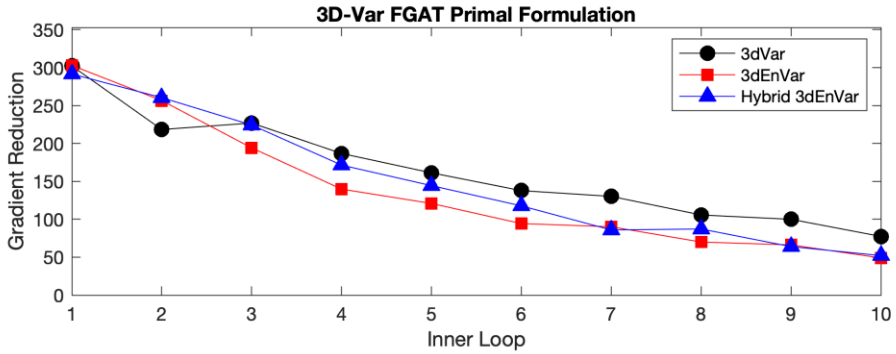
3D-Var Convergence: Gradient Reduction

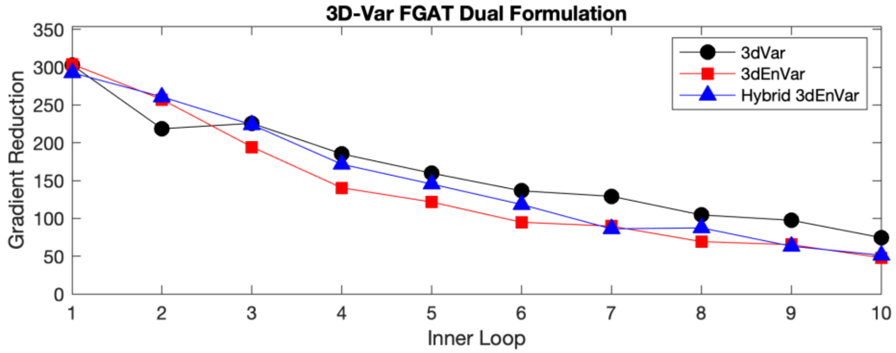
Regular (Primal/Dual)





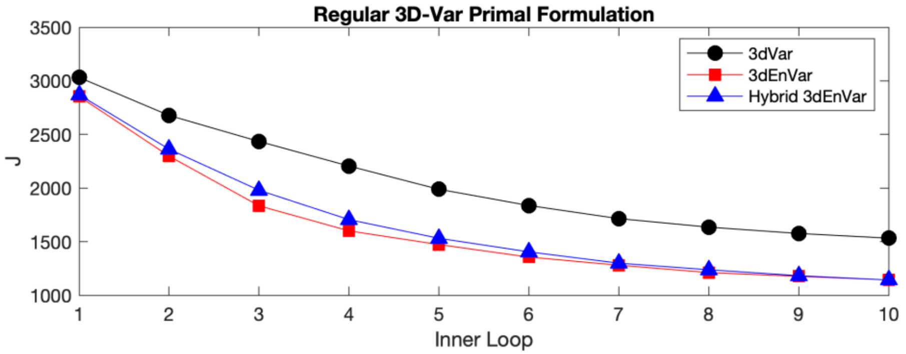
FGAT (Primal/Dual)

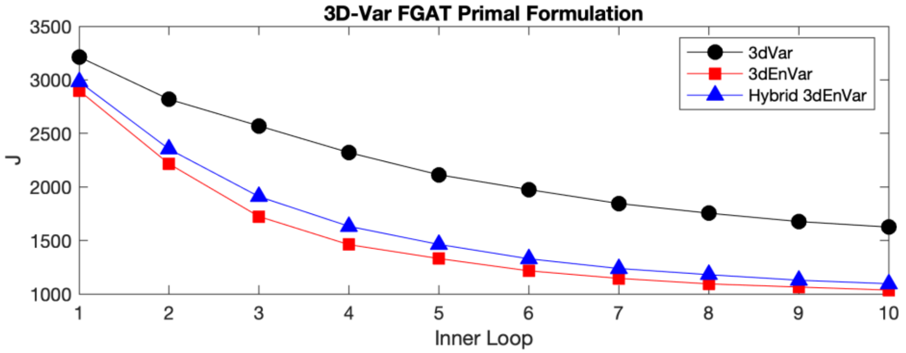




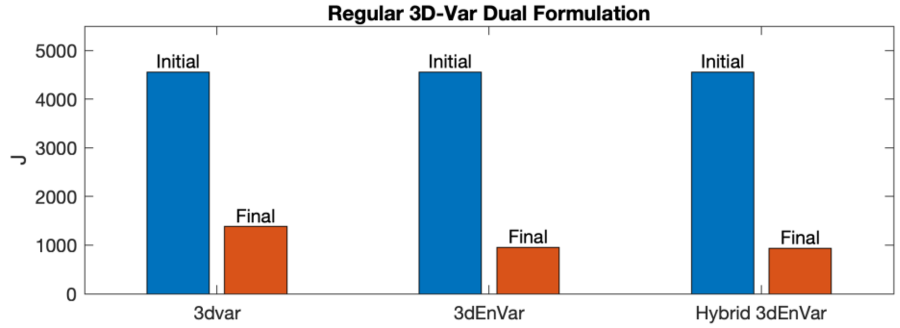
3D-Var Convergence: Nonlinear Cost Function

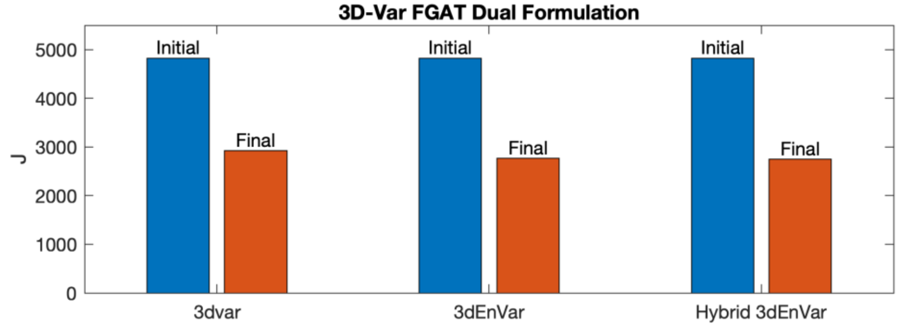
Primal Formulation





Dual Formulation





Implementing ROMS DA in JEDI Summary

- **Repositories created for ROMS-JEDI development within JCSDA:**
  - git clone https://github.com/JCSDA-internal/roms_src.git (latest **ROMS** source code)
  - git clone https://github.com/JCSDA-internal/roms-jedi.git (ongoing **ROMS-JEDI** interface)
- **ROMS-WC13 test case (~35-km U.S. West Coast) is configured as a realistic ocean application for interface Unit Tests.**
- All the elemental **C++** and **Fortran-2003** building blocks or object Classes (**OOPS**, **SABER**, **UFO,** and **VADER**) for the **ROMS-JEDI** interface have been coded and tested with **WC13**. Currently, there are **42 Unit Tests**.
- The **ROMS-JEDI** interface is up-to-date with the evolving changes in the **JEDI** building blocks.
- All the **DA** observation operators **H(x)** and **H^T(x)** use the **generic** **UFO** horizontal and vertical interpolation methods. ROMS depths are **negative** with **levelsAreTopDown()=false**. That is **bottom** (**k=1**) and **top** (**k=N**).
- The **ROMS-JEDI** Background Error Covariance **B** (static, ensemble, and hybrid) is modeled with **BUMP/NICAS**.
- We started testing and evaluating various variational data assimilation algorithms in **JEDI** for **3D-Var** and will progress to **4D-Var** using the primal and dual formulation with different minimization solvers.
- The **ROMS-JEDI** interphase was challenging and time-consuming to build and test, but it is incredibly trivial and easy to run once the user has all the observations stored in IODA NetCDF-4 files.

ROMS-JEDI 3D-Var FGAT Configuration: 3dvar_fgat_primal.yaml

```yaml
forecast length: &ForecastLength PT96H          # 96 hours (4 days) or smaller interval
model timestep:  &TimeStep PT1800S              # WC13 has a 30 minutes timestep
initial date: &date 2004-01-03T00:00:00Z        # initial DA window

roms analysis:  &roms_analysis [ssh, uocn, vocn, tocn, socn]
roms state:     &roms_state    [ssh, uocn, vocn, tocn, socn, zocn_r]

cost function:                                  # OOPS cost function type
  cost type: 4D-Var                             # using 4D-Var because FGAT
  window begin: *date
  window length: *ForecastLength                # 4-day DA cycle
  analysis variables: *roms_analysis
  geometry: &roms_geom1                         # ROMS application geometry
    project_dir: Data/wc13
    roms_stdinp: roms_wc13_20040103.in
    fields metadata: Data/fields_metadata.yaml
    ng: 1                                       # ROMS nested grid number
  model:                                        # ROMS nonlinear model object
    name: ROMS
    model variables: *roms_state
    simulation length: *ForecastLength
    tstep: *TimeStep
  background:                                   # ROMS state background / first guess
    fields_dir: jediroms_wc13/Data/
    fields_filename: wc13_roms_ini_20040103.nc  # initial fields for 2004-01-03T00:00:00Z
    fields_record: 1
    state variables: *roms_state
    date: *date
  background error:                             # background error covariance modeling
    covariance model: SABER
    saber central block:
      saber block name: BUMP_NICAS              # static background error covariance
      read:
        drivers:
          multivariate strategy: univariate
          read local nicas: true
        grids:
        - model:                                # 2D-fields correlations
            variables:
            - ssh                               # sea_surface_height
          io:
            files prefix: Data/bump/wc13_bump2d
        - model:                                # 3D fields correlations
            variables:
            - tocn                              # waterTemperature
            - socn                              # salinity
          io:
            files prefix: Data/bump/wc13_bump3d
    saber outer blocks:
    - saber block name: StdDev                  # ROMS state vector standard deviation
      read:
        model file:
          fields_dir: Data/wc13
          fields_filename: wc13_roms_std_i.nc   # same as ROMS native RBL4D-Var
          fields_record: 1
          state variables: *roms_analysis
          date: *date
```

```yaml
 observations:                                        # Observations to assimilate
   observers:
   - obs space:
       name: InsituTemperature
       obsdatain:
         engine:
           type: H5File
           obsfile: Data/obs/wc13_temp_20040103.nc4
       obsdataout:
         engine:
           type: H5File
           obsfile: Data/3dvar/fgat/primal/wc13_temp_3dvar_fgat.nc4
       simulated variables: [waterTemperature]
     obs operator:
       name: VertInterp
       vertical coordinate: model_level_depth_at_cell_center
       observation vertical coordinate: depth   # negative, levelsAreTopDown()=false, (bottom k=1)
       observation alias file: testinput/obsop_name_map.yaml
     obs error:
       covariance model: diagonal
# . . . . . . . . . . . . . . . .                     And so on for InsituSalinity, SST, and ADT
variational:                                          # minimization solver (Derber-Rosati Precond. Lanczos)
  minimizer:
    algorithm: DRPLanczos                             # Primal Formulation
  iterations:                                         # each item defines an outer loop
  - ninner: 10                                        # number of inner loops
    gradient norm reduction: 1.0e-10
    geometry: *roms_geom1                             # inner loops geometry object (can be coarser)
    linear model:
      name: Identity                                  # Identity TLM/ADM
      tstep: *TimeStep
      increment variables: *roms_analysis
      variable change: Identity
    diagnostics:                                      # Written into output IODA files
      departures: ombg                                # observation minus background: y - H(Xb)
    online diagnostics:
      write increment: true
      increment:                                      # DA increment output file
        single_record: true
        data_dir: Data/3dvar/fgat/primal
        prefix: wc13_roms
        exp: 3dvar
        type: inc
        date: *date
final:
  diagnostics:                                        # Written into output IODA files
    departures: oman                                  # observation minus analysis: y - H(Xa)
output:
  file_policy: PT98H                                  # single output file with 49 records
  frequency: &frequency PT2H                          # write output NLM fields every 2 hours
  data_frequency: *frequency
  data_dir: Data/3dvar/fgat/primal
  prefix: wc13_roms
  exp: 3dvar
  type: nlm
  date: *date
```

ROMS-JEDI

Acknowledgments:

- Many thanks to JEDI developers at JCSDA for developing such a fantastic generic Data Assimilation Framework.
- Many thanks to original OOPS developers at ECMWF.
- We thank the following colleagues for their kind help and for answering our software questions during the ROMS-JEDI interface development:
  - Anna Shlyaeva (JCSDA)
  - Benjamin Ménétrier
  - Guillaume Vernieres (NOAA)
  - Travis Sluka (JCSDA)
  - Yannick Tremolet (JCSDA)
- I'm grateful to Marcin Chrust (ECMWF) for explaining how OOPS works.
- Finally, We are grateful to NOAA for its continued support.

Thank you!