Ensemble Data Assimilation in NEMO using PDAF

Lars Nerger[1], Yuchen Sun[1], Wibke Düsterhöft-Wriggers[2],
Yumeng Chen[3], and Dale Partridge[4]

1: Alfred Wegener Institute, Bremerhaven, Germany
2: Federal Maritime and Hydrographic Agency (BSH), Rostock, Germany
3: University of Reading / NCEO, Reading, UK
4: Plymouth Marine Laboratory, Plymouth, UK

PDAF Parallel Data Assimilation Framework

This work has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101004032.

PDAF – Parallel Data Assimilation Framework

A unified tool for interdisciplinary data assimilation …

- provide support for parallel ensemble forecasts
- provide assimilation methods (solvers) - fully-implemented & parallelized
- provide tools for observation handling and for diagnostics
- easily useable with (probably) any numerical model
- a program library (PDAF-core) plus additional functions
- run from notebooks to supercomputers (Fortran, MPI & OpenMP)
- ensure separation of concerns (model – DA method – observations – covariances)

Open source:
Code, documentation, and tutorial available at

https://pdaf.awi.de

github.com/PDAF

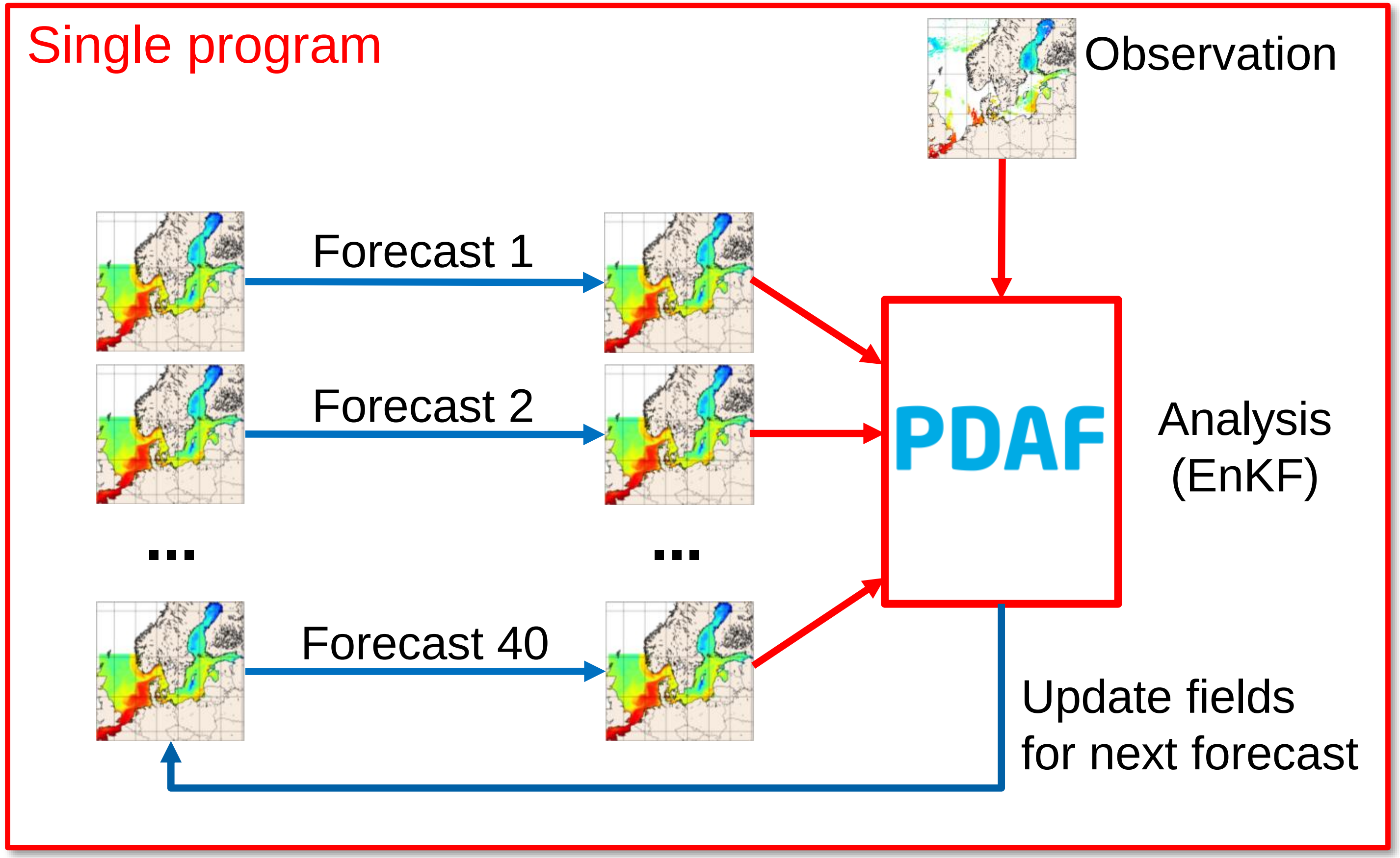
L. Nerger, W. Hiller, Computers & Geosciences 55 (2013) 110-118

Online-Coupling – Assimilation-enabled Model

Couple a model with PDAF

- Modify model to simulate ensemble of model states
- Insert analysis step/solver to be executed at prescribed interval
- Run model as usual, but with more processors and additional options

- EnOI and 3D-Var also possible:
  - Evolve single model state
  - Prescribe ensemble perturbations or covariance



Initialize ensemble

Ensemble forecast

Analysis step in between time steps

NEMO-PDAF Online Coupling

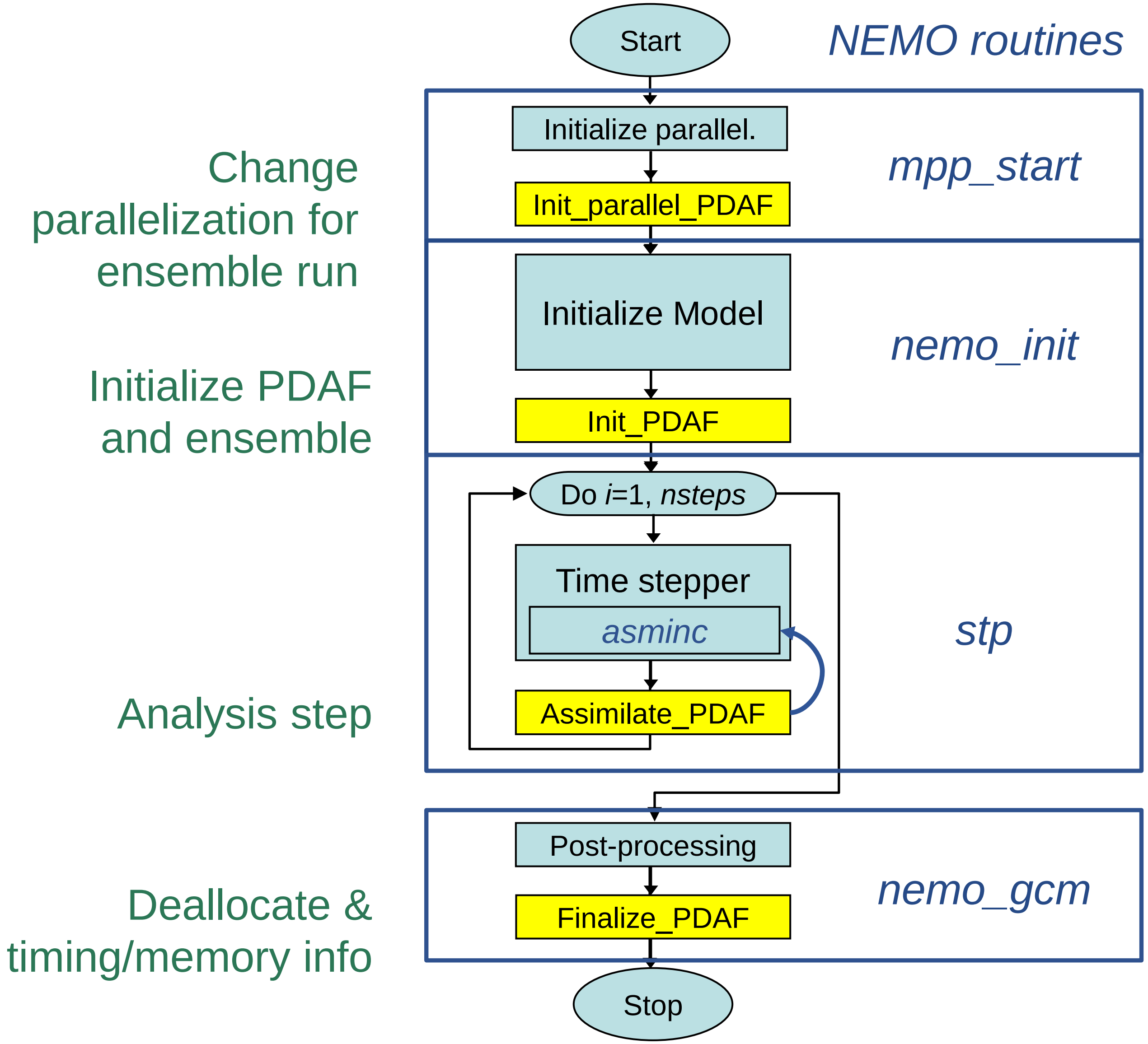
Model code | DA code

PDAF Parallel Data Assimilation Framework

Augment NEMO with DA functionality

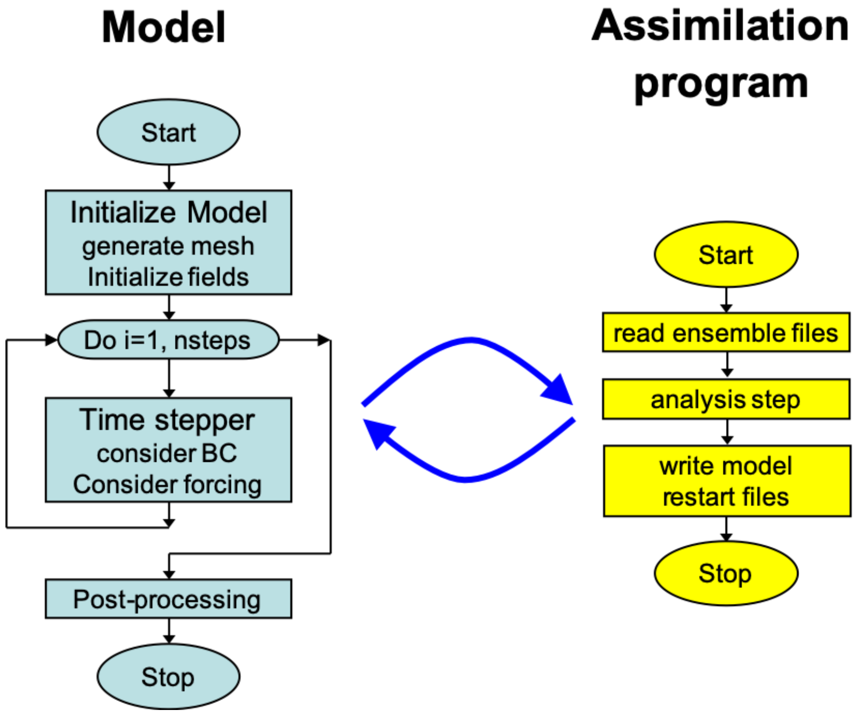
- Insert 4 subroutine calls
- Modify parallelization for ensemble mode
- Transfer data between model and PDAF in memory
- Adapt NEMO's ASM module:
  PDAF provides increments in memory
- No changes to XIOS, but to file_def*.xml

- Very efficient & highly scalable
- Direct access to model grid and fields through Fortran modules
- File output kept to minimum
- No model restarts for cycled assimilation



NEMO-PDAF Offline Coupling

Offline



Currently used by CMEMS Baltic-MFC with NEMO-4

- Separate programs for model and DA
  - no modification of model code (unless e.g. model error is to be applied)
- Flexible to run
  - each model task can be run separately
- Needs frequent model restarts and file output
  - overhead by model restarts
  - overhead by file writing & reading

Assimilation program

- mostly same routines as for online-coupling
- differences
  - reading restart and writing increment files
  - get information on model grid from files (at central point in code)
  - compression of state vector to wet grid points

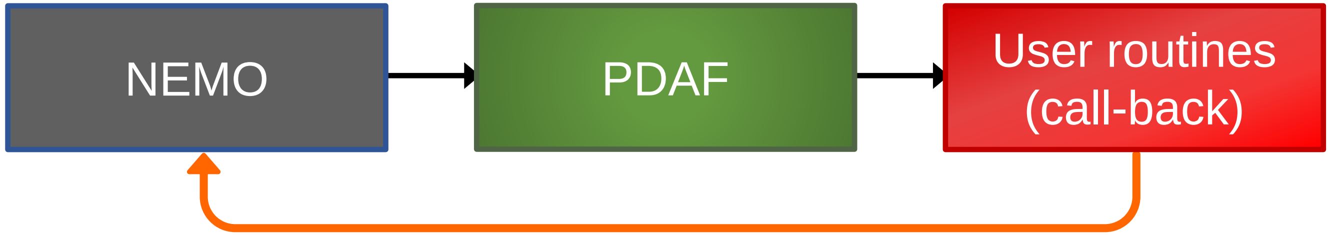
Model code | DA code

PDAF interface structure

- **Model-sided Interface:** Defined calls to PDAF routines
  (called by driver program for offline coupling)
- **Case-related Interface:**
  User-supplied call-back routines for elementary operations:
  - transfers between model fields and ensemble of state vectors
  - observation-related operations
- User-supplied routines can be implemented as routines of the model and can share data with it (low abstraction level)

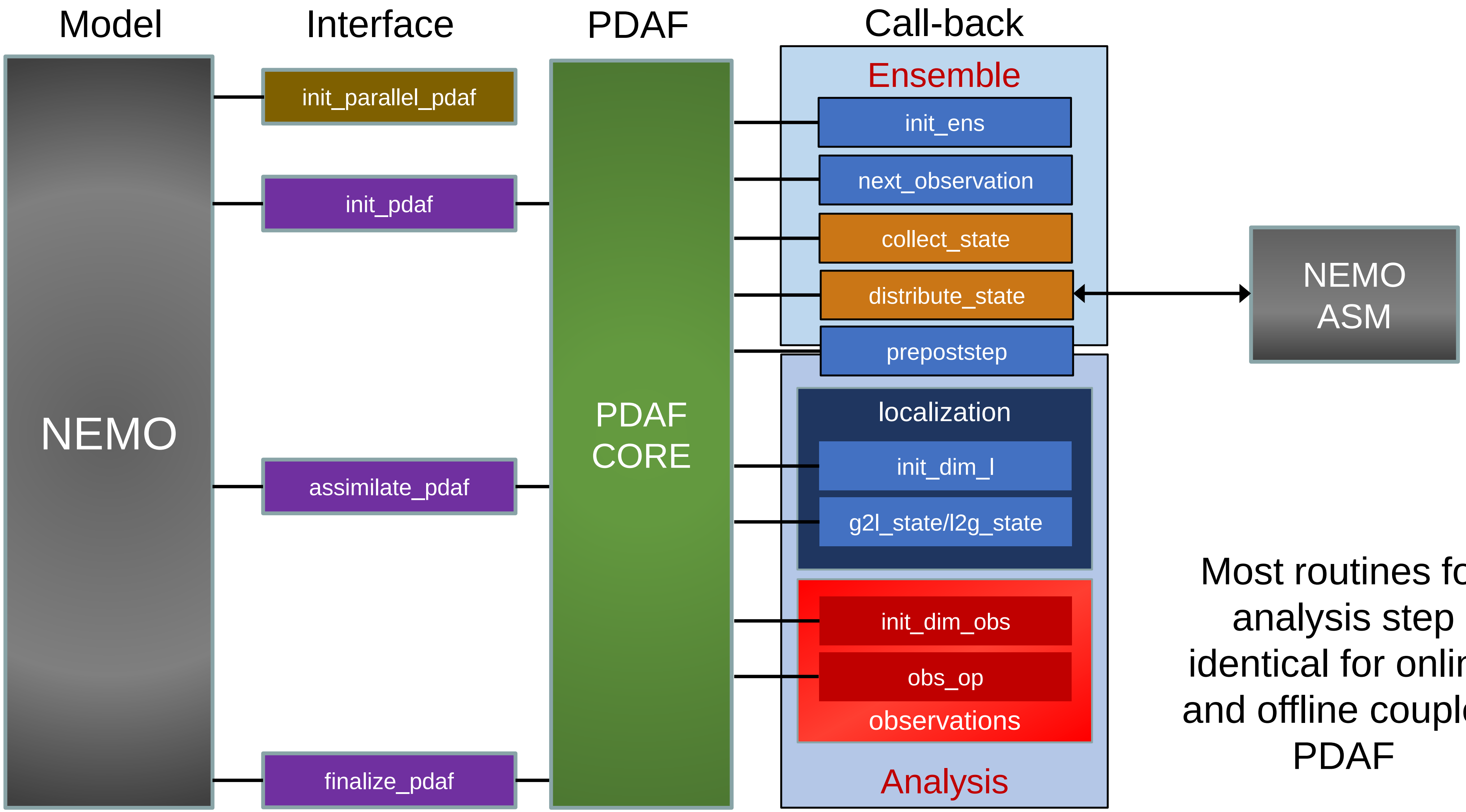
PDAF is model agnostic

model specific code only in user-supplied routines



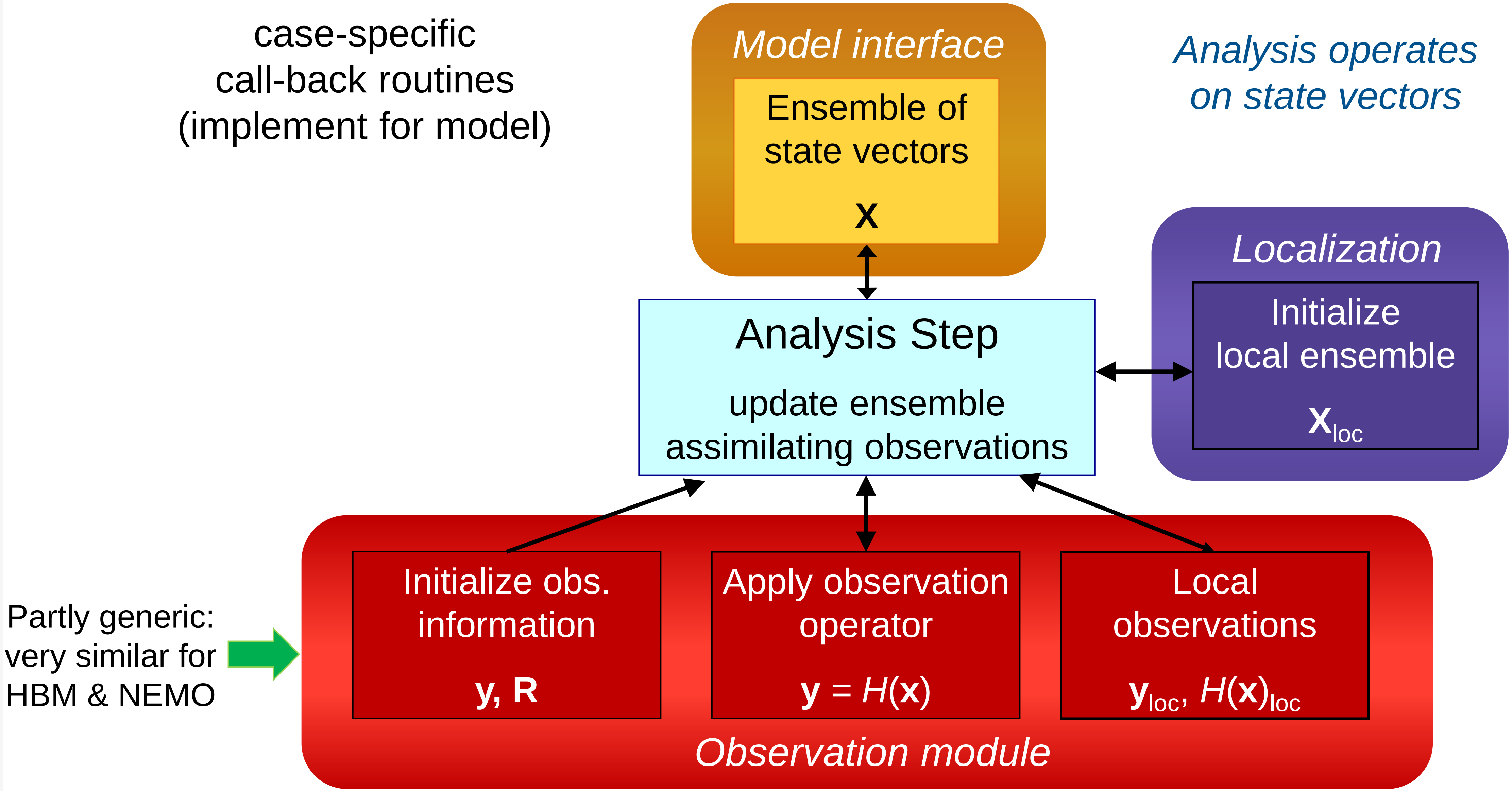
Online: Access model information through modules / static variables
offline: initialize model information from files

PDAF Calling Structure



Most routines for analysis step identical for online and offline coupled PDAF

Implementing the Ensemble Filter Analysis Step



case-specific
call-back routines
(implement for model)

Model interface

Ensemble of state vectors

$\mathbf{X}$

Analysis operates on state vectors

Localization

Initialize local ensemble

$\mathbf{X}_{\mathrm{loc}}$

Analysis Step

update ensemble assimilating observations

Partly generic: very similar for HBM & NEMO

Initialize obs. information

$\mathbf{y}, \mathbf{R}$

Apply observation operator

$\mathbf{y} = H(\mathbf{x})$

Local observations

$\mathbf{y}_{\mathrm{loc}}, H(\mathbf{x})_{\mathrm{loc}}$

Observation module

Adaptions to ASMINC

- Settings for asminc provided by PDAF user code
    - Ensure consistency between PDAF and ASM

- Adaptions for cycled DA
    - Adapt IAU weights array for repeated application
    - Direct initialization option performed in PDAF user code
    - PDAF user-code initializes bkginc and bgk fields
    - Divergence damping routine in user code for repeated application

Assimilating Phytoplankton Carbon in NEMO-FABM-MEDUSA

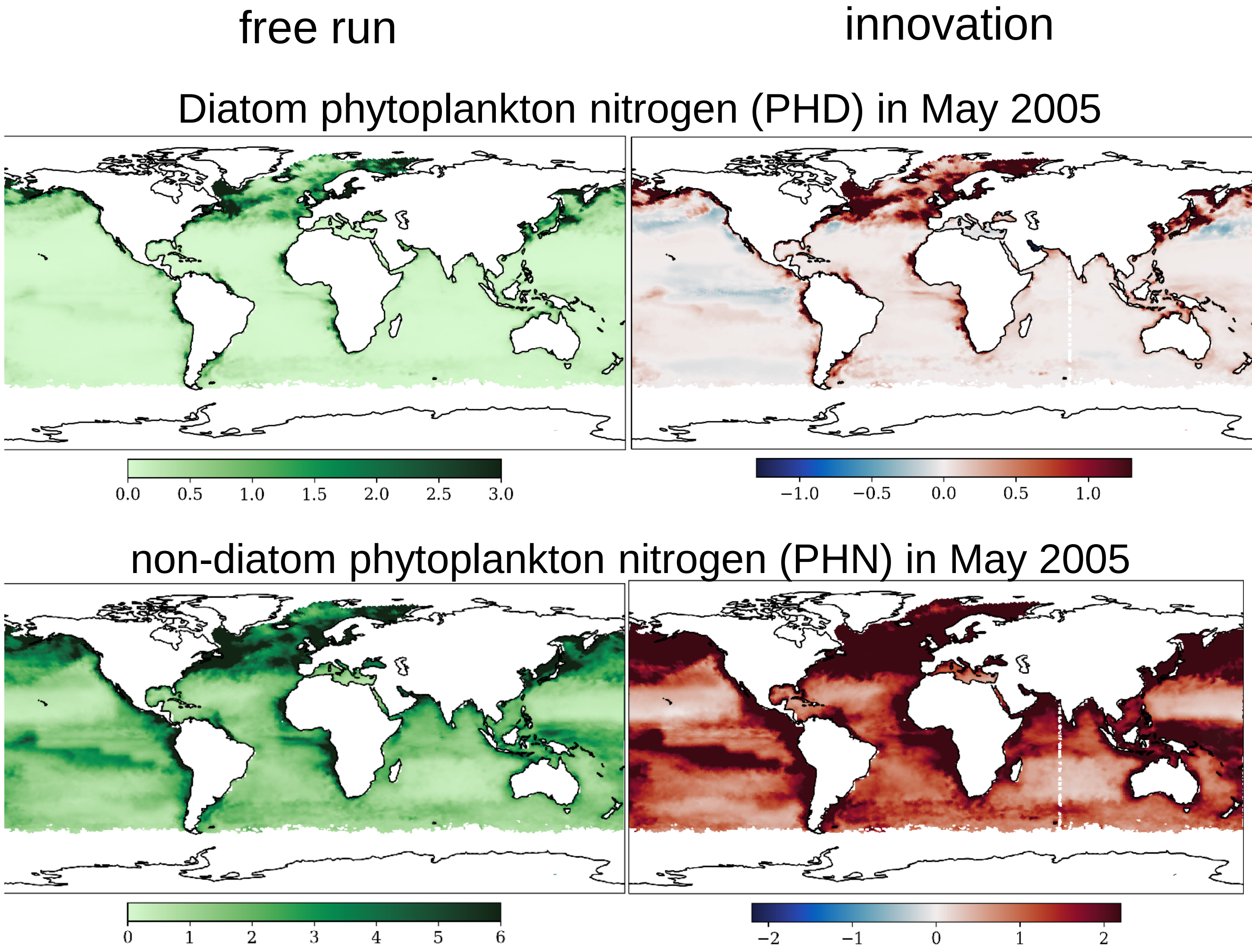
Model setup

- ORCA1 grid
- Year: 2005
- Base currency: nitrogen

Observations

- phytoplankton carbon products from BICEP project
- gridded product with a resolution of 9 km
- superobbing around 0.5°
- conversion from carbon to nitrogen based on a fixed C:N ratio specified by the model

→ Biased non-diatom phytoplankton nitrogen


free run

innovation

Diatom phytoplankton nitrogen (PHD) in May 2005

non-diatom phytoplankton nitrogen (PHN) in May 2005

National Centre for Earth Observation
NATURAL ENVIRONMENT RESEARCH COUNCIL

University of Reading

PML Plymouth Marine Laboratory

Assimilating Phytoplankton Carbon in NEMO-FABM-MEDUSA

DA setup

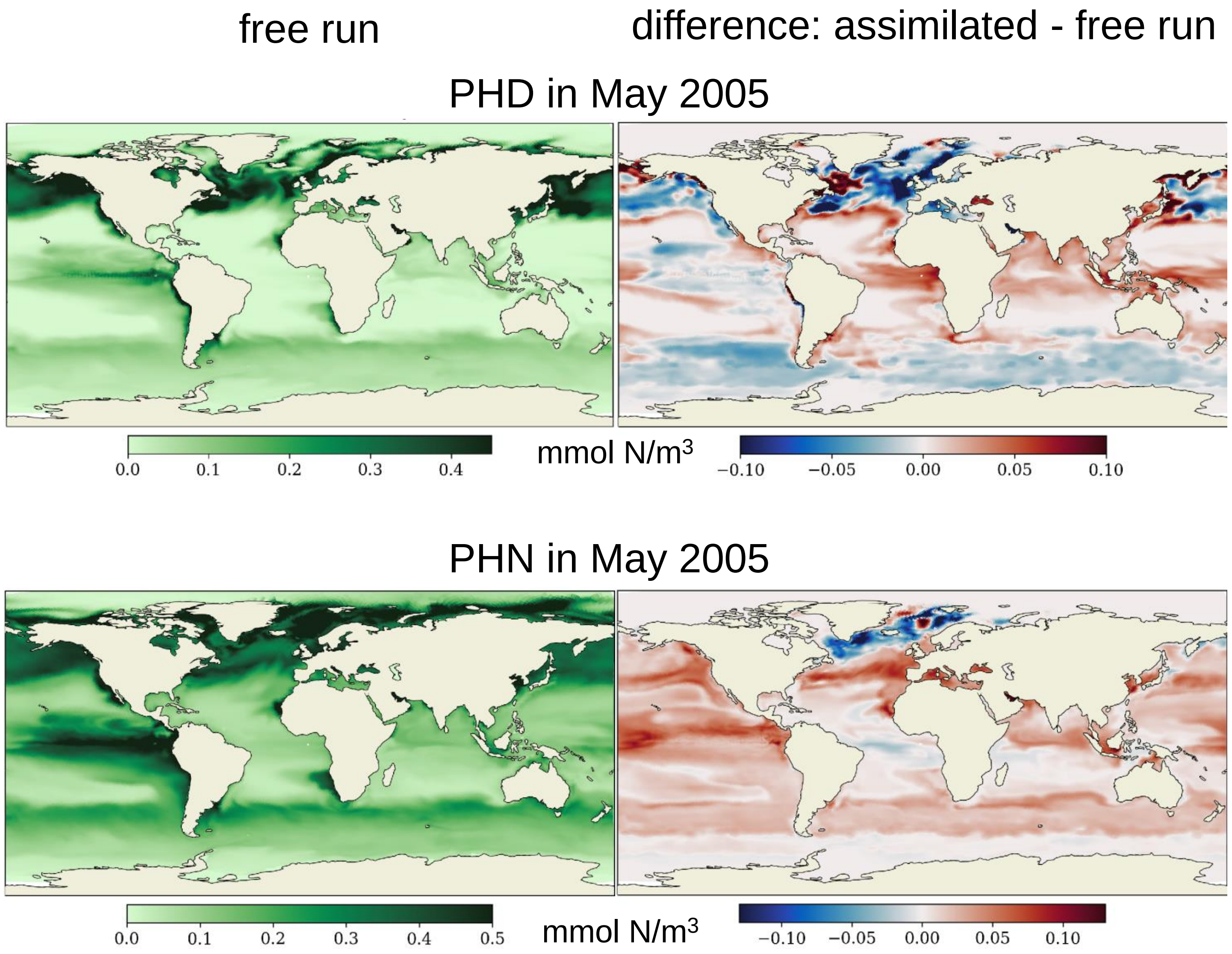
- Kalman filter: LESTKF
- Localisation radius: 200 km
- Forgetting factor: 0.95
- Ensemble size: 30

Technicalities

- No. PEs for ensemble models: 5400
- No. XIOS servers: 480
- Runtime for one-year: ~9 hrs
- Slower than individual ensemble run
- Potential Bottleneck:
  - I/O: switching to multiple file output in the hope for faster I/O

State vector

- Diatom phytoplankton nitrogen (PHD)
- non-diatom phytoplankton nitrogen (PHN)
- Dimension of the state vector: 1.45×10^7



free run

difference: assimilated - free run

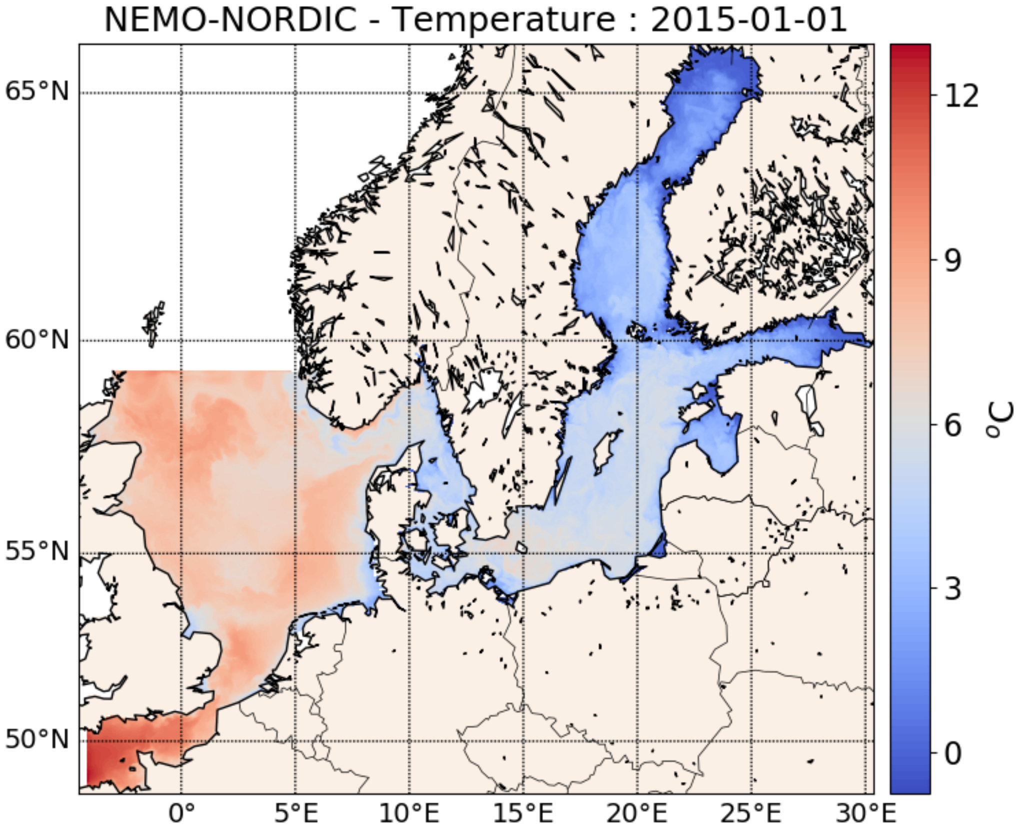
PHD in May 2005

PHN in May 2005

Assimilating SST and Chlorophyll in NEMO-ERGOM

- Model setup
  - NEMO-NORDIC + ERGOM biogeochemistry
  - 1 nm resolution (ca. 1.8km); 56 layers
  - time step 90 sec
- DA setup
  - Kalman filter: LESTKF
  - Ensemble size: 30
- Observations
  - SST and chlorophyll from CMEMS
- State
  - 5 physics variables
  - 16 ERGOM prognostic variables + 4 diagnostic variables
  - state dimension: $704 \cdot 10^6$ (at analysis $153 \cdot 10^6$)
- Compute requirements:
  - each NEMO-ERGOM task: 186 cores (+6 for XIOS)
  - run times: forecasts ~45 minutes per 24h
    analysis 0.5 minutes
    file writing 3 minutes (2x state, 1x variance)

Results to be presented in my talk tomorrow



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004032.

Some Challenges

- Initial run problems
  - Program got stuck when integration started
    (apparently in reading of forcing and boundary data)
- Needed to
  - Convert all input files to NetCDF4
  - Transition from Intel MPI to OpenMPI 4
    (issue of combination of XIOS/MPI/parallel NetCDF)
  - Use distributed restart file instead of global ones

PDAF package: DA Algorithms and Models

PDAF originated from comparison studies of different filters

Ensemble Filters and smoothers - *global and localized*

- EnKF (Evensen, 1994, perturbed obs.)
- (L)ETKF (Bishop et al., 2001/Hunt et al. 2007)
- (L)ESTKF (Nerger et al., 2012)
- (L)NETF (Toedter & Ahrens, 2015)
- Particle filter
- Hybrid LETKF-LNETF
- *EnOI mode*

Model bindings

- MITgcm
- AWI-CM / FESOM

Toy models (full implementations with PDAF)

- Lorenz-96 / Lorenz-63
- Lorenz-2005 models II and III

User routines not yet coded for NEMO

3D-Var schemes
(incremental with control variable transformation)

- 3D-Var with parameterized covar.
- 3D Ensemble Var
- Hybrid 3D-Var

Community:

- pyPDAF (Python-coded models)
- TerrSysMP-PDAF

In progress

- SCHISM/ESMF (VIMS)
- GOTM/FABM “EAT” (BB ApS)

Summary – NEMO-PDAF

- Coupled NEMO and PDAF for (ensemble) data assimilation
  - state vector can include ocean physics / sea ice / BGC …
  - easy addition of observation types
  - supports cycled DA without model restarts
  - utilize NEMO’s ASM-module: IAU and direct initialization
  - Currently for NEMO 4.0.x – NEMO 4.2 in progress

- Code will be made open source

PDAF is open source:
Code, documentation, and tutorial available at

https://pdaf.awi.de

github.com/PDAF

References

- https://pdaf.awi.de (The website also provides a list of studies using PDAF)
- https://github.com/PDAF
- Nerger, L., Hiller, W. (2013). Software for Ensemble-based Data Assimilation Systems - Implementation Strategies and Scalability. Computers and Geosciences, 55, 110-118. doi:10.1016/j.cageo.2012.03.026
- Nerger, L., Hiller, W., Schröter, J.(2005). PDAF - The Parallel Data Assimilation Framework: Experiences with Kalman Filtering, Use of high performance computing in meteorology : proceedings of the Eleventh ECMWF Workshop on the Use of High Performance Computing in Meteorology, Reading, UK, 25 - 29 October 2004 / Eds.: Walter Zwieflhofer; George Mozdzynski, Singapore: World Scientific, 63-83. doi:10.1142/9789812701831_0006
- Nerger, L., Tang, Q., Mu, L. (2020). Efficient ensemble data assimilation for coupled models with the Parallel Data Assimilation Framework: Example of AWI-CM. Geoscientific Model Development, 13, 4305–4321, doi:10.5194/gmd-13-4305-2020