Building Ensemble-Based Data Assimilation Systems for Coupled Models

Lars Nerger
Alfred Wegener Institute for Polar and Marine Research
Bremerhaven, Germany
Overview

How to simplify to apply data assimilation?
1. Extend model to integrate the ensemble
2. Add analysis step to the model
3. Then focus on applying data assimilation
PDAF: A tool for data assimilation

PDAF - Parallel Data Assimilation Framework

- a program library for ensemble data assimilation
- provide support for parallel ensemble forecasts
- provide fully-implemented & parallelized filters and smoothers (EnKF, LETKF, NETF, EWPF … easy to add more)
- easily useable with (probably) any numerical model (applied with NEMO, MITgcm, FESOM, HBM, TerrSysMP, …)
- run from laptops to supercomputers (Fortran, MPI & OpenMP)
- first public release in 2004; continued development
- ~200 registered users; community contributions

Open source:
Code, documentation & tutorials at
http://pdaf.awi.de

Application examples run with PDAF

- FESOM: Global ocean state estimation (Janjic et al., 2011, 2012)
- HBM-ERGOM: Coastal assimilation of SST & ocean color (S. Losa et al. 2013, 2014)
- MITgcm: sea-ice assimilation (Q. Yang et al., 2014-16, NMEFC Beijing)

+ external applications & users, e.g.
  - Geodynamo (IPGP Paris, A. Fournier)
  - MPI-ESM (coupled ESM, IFM Hamburg, S. Brune) -> talk tomorrow
  - CMEMS BAL-MFC (Copernicus Marine Service Baltic Sea)
  - TerrSysMP-PDAF (hydrology, FZJ)
Ensemble filter analysis step

Analysis operates on state vectors (all fields in one vector)

Filter analysis
1. update mean state
2. ensemble transformation

Ensemble of state vectors
\( X \)

Vector of observations
\( y \)

Observation operator
\( H(...) \)

Observation error covariance matrix
\( R \)

For localization:
Local ensemble
Local observations

case-specific call-back routines
Logical separation of assimilation system

- **Ensemble Filter**
  - Initialization
  - Analysis
  - Ensemble transformation

- **Model**
  - Initialization
  - Time integration
  - Post processing

- **Observations**
  - Quality control
  - Obs. vector
  - Obs. operator
  - Obs. error

- **Single program**
  - State
  - Time

- **Core of PDAF**
  - Mesh data

- **Explicit interface**

- **Indirect exchange (module/common)**

---

Extending a Model for Data Assimilation

Model
- **single or multiple executables**
- **coupler might be separate program**

Extended parallelization enables ensemble forecast

Extension for data assimilation

---

Start
- Initialize parallel.
- Initialize Model
  - Initialize coupler
  - Initialize grid & fields
- Do $i=1$, nsteps
  - Time stepper
  - in-compartment step coupling
- Post-processing
- Stop

Init_PDAF
- Do $i=1$, nsteps
  - Time stepper
  - in-compartment step coupling
- Assimilate_PDAF
- Post-processing
- Stop

Init_parallel_PDAF
- Initialize Model
  - Initialize coupler
  - Initialize grid & fields
- Init_parallel_PDAF
- Possible model-specific adaption
- Possible adaption of coupler (e.g. OASIS3-MCT)

---

Lars Nerger et al. – Building EnsDA Systems for Coupled Models
Framework solution with generic filter implementation

Start
- init_parallel_DA
  - Initialize Model
    - Init_DA
      - Do $i=1, nsteps$
        - Time stepper
          - Assimilate
            - Post-processing
              - Stop

Subroutine calls or parallel communication
*No files needed!*

PDAF_Init
  - Set parameters
  - Initialize ensemble

PDAF_Analysis
  - Check time step
  - Perform analysis
  - Write results

PDAF_Init
  - Read ensemble from files
  - Initialize state vector from model fields
  - Initialize vector of observations
  - Apply observation operator to a state vector
  - multiply R-matrix with a matrix

Model with assimilation extension
Core-routines of assimilation framework
Case specific callback routines

Lars Nerger et al. – Building EnsDA Systems for Coupled Models
2-level Parallelism

1. Multiple concurrent model tasks
2. Each model task can be parallelized
   - Analysis step is also parallelized

- Configured by “MPI Communicators”

Lars Nerger et al. – Building EnsDA Systems for Coupled Models
Problem reduces to:

1. Configuration of parallelization (MPI communicators)

2. Implementation of compartment-specific user routines and linking with model codes at compile time
Lars Nerger et al. – Building EnsDA Systems for Coupled Models
Configure Parallelization – weakly coupled DA

Logical decomposition:
- Communicator for each
  - Coupled model task
    - Compartment in each task (init by coupler)
      - (Coupler might want to split MPI_COMM_WORLD)
  - Filter for each compartment
  - Connection for collecting ensembles for filtering
- Different compartments
  - Initialize distinct assimilation parameters
  - Use distinct user routines
Example: TerrSysMP-PDAF (Kurtz et al. 2016)

TerrSysMP model
- Atmosphere: COSMO
- Land surface: CLM
- Subsurface: ParFlow
- coupled with PDAF using wrapper
- single executable
- driver controls program
- Tested using 65536 processor cores
Example: ECHAM6-FESOM

Atmosphere
- ECHAM6
- JSBACH land

Ocean
- FESOM
- includes sea ice

Coupler library
- OASIS3-MCT

Separate executables for atmosphere and ocean

Data assimilation (FESOM completed, ECHAM6 in progress)
- Add 3 subroutine calls per compartment model
- Replace MPI_COMM_WORLD in OASIS coupler
- Implement call-back routines

Model: D. Sidorenko et al., Clim Dyn 44 (2015) 757
Summary

• Software framework simplifies building data assimilation systems
• Efficient online DA coupling with minimal changes to model code
• Setup of data assimilation with coupled model
  1. Configuration of communicators
  2. Implementation of user-routines
     • for interfacing with model code and
     • observation handling
References

- http://pdaf.awi.de


Thank you!

Lars.Nerger@awi.de - Building EnsDA Systems for Coupled Models