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

Vector of observations $y$
Observation operator $H(\cdot)$
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

Core of PDAF

- Model
  - initialization
  - time integration
  - post processing

- Observations
  - quality control
  - obs. vector
  - obs. operator
  - obs. error

single program

Explicit interface

Indirect exchange (module/common)

Extending a Model for Data Assimilation

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

revised parallelization enables ensemble forecast

Extension for data assimilation

plus:
- Possible model-specific adaption
- Possible adaption of coupler (e.g. OASIS3-MCT)
Framework solution with generic filter implementation

Start

*init_parallel_DA*

**Initialize Model**

*Init_DA*

Do $i=1, nsteps$

**Time steppper**

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

Dependent on model and observations

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 call-back routines

Lars Nerger et al. – Building EnsDA Systems for Coupled Models
1. Multiple concurrent model tasks
2. Each model task can be parallelized
   - Analysis step is also parallelized
   - Configured by “MPI Communicators”
Building the Assimilation System

Problem reduces to:

1. Configuration of parallelization (MPI communicators)

2. Implementation of compartment-specific user routines and linking with model codes at compile time
2 compartment system – strongly coupled DA

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
Changes to FESOM

Add to `par_init` (gen_partitioning.F90) after `MPI_init`

```fortran
#ifdef USE_PDAF
   CALL init_parallel_pdaf(0, 1, MPI_COMM_FESOM)
#endif
```

Add to `main` (fesom_main.F90) just before stepping loop

```fortran
CALL init_pdaf()
```

Add to `main` (fesom_main.F90) just before ‘END DO’

```fortran
CALL assimilate_pdaf()
```

OASIS3-MCT

Assumes to split `MPI_COMM_WORLD` in `oasis_init_comp` (mod_oasis_method.F90)

- Needs to split `COMM_FESOM`
Changes to ECHAM6

Add to $p_{\text{start}}$ (mo_mpi.f90) after MPI_init

```c
#ifdef USE_PDAF
    CALL init_parallel_pdaf(0, 1, p_global_comm)
#endif
```

Add to control (control.f90) before call to stepon

```c
CALL init_pdaf()
```

Add to stepon (step.f90) before ‘END DO’

```c
CALL assimilate_pdaf()
```

OASIS3-MCT

Assumes to split MPI_COMM_WORLD in oasis_init_comp (mod_oasis_method.F90)

- Needs to split p_global_comm
Minimal changes to NEMO

Add to *mynode* (lin_mpp.F90) just before init of myrank

```fortran
#ifdef key_USE_PDAF
   CALL init_parallel_pdaf(0, 1, mpi_comm_opa)
#endif
```

Add to *nemo_init* (nemogcm.F90) at end of routine

```fortran
CALL init_pdaf()
```

Add to *stp* (step.F90) at end of routine

```fortran
CALL assimilate_pdaf()
```

For Euler time step after analysis step:

Modify *dyn_nxt* (dynnxt.F90)

```fortran
#ifdef key_USE_PDAF
   IF((neuler==0 .AND. kt==nit000) .OR. assimilate)
#else
Lars Nerger et al. – Building EnsDA Systems for Coupled Models
PDAF originated from comparison studies of different filters

Filters
- EnKF (Evensen, 1994 + perturbed obs.)
- ETKF (Bishop et al., 2001)
- SEIK filter (Pham et al., 1998)
- SEEK filter (Pham et al., 1998)
- ESTKF (Nerger et al., 2012)
- LETKF (Hunt et al., 2007)
- LSEIK filter (Nerger et al., 2006)
- LESTKF (Nerger et al., 2012)

Smoothers for
- ETKF/LETKF
- ESTKF/LESTKF
- EnKF

Not yet released:
- serial EnSRF
- particle filter
- EWPF
- NETF

Global filters
Localized filters
Global and local smoothers

Systems for Coupled Models
Parallel Performance
Parallel Performance

Use between 64 and 4096 processor cores of SGI Altix ICE cluster (HLRN-II)

94-99% of computing time in model integrations

**Speedup**: Increase number of processes for each model task, fixed ensemble size

- factor 6 for 8x processes/model task
- one reason: time stepping solver needs more iterations

**Scalability**: Increase ensemble size, fixed number of processes per model task

- increase by ~7% from 512 to 4096 processes (8x ensemble size)
- one reason: more communication on the network
Very big test case

- Simulate a “model”
- Choose an ensemble
  - state vector per processor: $10^7$
  - observations per processor: $2 \cdot 10^5$
- Ensemble size: 25
- 2GB memory per processor
- Apply analysis step for different processor numbers
  - 12 – 120 – 1200 – 12000

- Very small increase in analysis time (~1%)
- Didn’t try to run a real ensemble of largest state size (no model yet)
Application examples run with PDAF

- FESOM: Global ocean state estimation (Janjic et al., 2011, 2012)
- HBM: Coastal assimilation of SST and in situ data (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)