Ensemble Data Assimilation

with the Parallel Data Assimilation Framework PDAF

Lars Nerger

Alfred Wegener Institute for Polar and Marine Research
Bremerhaven, Germany

Acknowledgments to:
Svetlana Losa, Paul Kirchgessner, Jens Schröter, Wolfgang Hiller,
Himansu Pradhan, Michael Goodliff, Qi Tang
Outline

- Implementation of data assimilation:
  - Parallel Data Assimilation Framework PDAF
- Application examples:
  - Regional ocean and ocean-biogeochemical data assimilation in the North and Baltic Seas
  - Coupled atmosphere-ocean model
Data Assimilation

Combine Models and Observations
Motivation

*Model* surface temperature

*Satellite* surface temperature

Information: Model

Information: Observations

Combine both sources of information quantitatively by computer algorithm

⇒ Data Assimilation

Data Assimilation

Combine model with real data

- Optimal estimation of system state:
  - initial conditions (for weather/ocean forecasts, ...)
  - state trajectory (temperature, concentrations, ...)
  - parameters (ice strength, plankton growth, ...)
  - fluxes (heat, primary production, ...)
  - boundary conditions and ‘forcing’ (wind stress, ...)

- More advanced: Improvement of model formulation
  - Detect systematic errors (bias)
  - Revise parameterizations based on parameter estimates
Implement Ensemble Data Assimilation

Parallel Data Assimilation Framework (PDAF)
Computational and Practical Issues

- Running a whole model ensemble is costly
- Ensemble propagation is naturally parallel (all independent)
- Ensemble data assimilation methods need tuning
- No need to go into model numerics (just model forecasts)
- Filter step of assimilation only needs to know:
  - Values of model fields and their location
  - Observed values, their location and uncertainty

Ensemble data assimilation can be implemented in form of a generic code
+ case-specific routines
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
- ~250 registered users; community contributions

Open source:
Code, documentation & tutorials at
http://pdaf.awi.de
Offline coupling – separate programs

Model

Start

Initialize Model
- generate mesh
- Initialize fields

Do i=1, nsteps

Time stepper
- consider BC
- Consider forcing

Post-processing

Stop

Assimilation program

Start

Read ensemble files

Analysis step

Write model restart files

Stop

generic

For each ensemble state
- Initialize from restart files
- Integrate
- Write restart files

- Read restart files (ensemble)
- Compute analysis step
- Write new restart files
Online-Coupling

single program

Ensemble Filter
Initialization
analysis
ensemble transformation

Core of PDAF

Model
initialization
time integration
post processing
modify parallelization

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

Explicit interface

Indirect exchange (module/common)

Extending a Model for Data Assimilation

**Model**

- single or multiple executables
- coupler might be separate program

**Extension for data assimilation**

- plus: Possible model-specific adaption
  - e.g. NEMO: Euler time step after assimilation

**revised parallelization enables ensemble forecast**

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

Start
- Initialize parallel.
- Initialize Model
  - Initialize coupler
  - Initialize grid & fields
- Init_parallel_PDAF
- Init_PDAF
- Do $i=1, nsteps$
  - Time stepper in-compartment step coupling
    - Assimilate_PDAF
  - Post-processing
    - Finalize_PDAF
- Stop
2-level Parallelism

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

- **Model Interface**
  - Ensemble of state vectors $X$

- **Filter analysis**
  - Update ensemble assimilating observations

Case-specific call-back routines

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

- **Observation Module**
  - For localization:
    - Local ensemble
    - Local observations

- **Vector of observations** $y$
- **Observation operator** $H(...)$
- **Observation error covariance matrix** $R$
User-supplied routines (call-back)

- Model und observation specific operations
- Elementary subroutines implemented in model context
- Called by PDAF routines though a defined interface
  - initialize model fields from state vector
  - initialize state vector from model fields
  - application of observation operator $H$ to some vector
  - initialization of vector of observations
  - multiplication with observation error covariance matrix
Framework solution with generic filter implementation

Subroutine calls or parallel communication

No files needed!

Model with assimilation extension

Core-routines of assimilation framework

Case specific callback routines

Lars Nerger

Ensemble Data Assimilation with PDAF
PDAF: Design

- Separate model developments from developments in data assimilation methods

- Efficiency:
  - direct online coupling of model and data assimilation method avoids frequent writing of ensembles to files
  - complete parallelism in model, filter, and ensemble integrations

- Simplified implementation:
  - minimal changes to model code when combining model with PDAF (extend model for data assimilation)
  - model not required to be a subroutine
  - control of assimilation program coming from model
  - simple switching between different filters and data sets

  ➢ Allows “users” to focus on their application
PDAF: User-friendliness

Assumption: Users know their model
→ let users implement DA system in model context

For users, model is not just a forward operator
→ let users extend their model for data assimilation

Keep simple things simple:

- Define subroutine interfaces to separate model and assimilation based on arrays
- No object-oriented programming (most models don’t use it; most model developers don’t know it; not many objects would be involved)
- Users directly implement observation specific routines (no indirect description of e.g. observation layout)
Application examples run with PDAF

- FESOM: Global ocean state estimation (Janjic et al., 2011, 2012)
- HBM: Coastal assimilation of SST, in situ and ocean color (S. Losa et al. 2013, 2014)
- MITgcm: sea-ice assimilation (Q. Yang et al., 2014-17, NMEFC Beijing)
- MITgcm-REcoM: ocean color assimilation
- AWI-CM: coupled atmos.-ocean assimilation

+ external applications & users, e.g.
  - Geodynamo (IPGP Paris, A. Fournier)
  - TerrSysMP-PDAF (hydrology, FZJ)
  - MPI-ESM (coupled ESM, IFM Hamburg, S. Brune)
  - CMEMS BAL-MFC (Copernicus Marine Service Baltic Sea)
  - CFSv2 (J. Liu, IAP-CAS Beijing)
Parallel Performance (FESOM-PDAF)

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 Example

Assimilation in the North and Baltic Seas
Grid nesting:
- 10 km grid
- 5 km grid
- 900 m, 25 layers

Operational BSH Model – BSHcmod, now HBM

10 km grid is used offline as boundary condition.

Longer cooperation BSH-AWI:
- Use operational model of BSH (Federal Maritime and Hydrographic Agency)
- Improve forecast skill of operational model using ensemble data assimilation
- Test system pre-operationally
- Extend assimilation to biogeochemical model ERGOM
Observations

- sea surface temperature from NOAA satellites
- 12-hour composites
- Interpolated to both model grids
- Observation error: 0.8 °C

## Configuration for BSHcmod data assimilation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting/Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter</td>
<td>Local SEIK</td>
</tr>
<tr>
<td>Ensemble size</td>
<td>8 members (trial and error)</td>
</tr>
<tr>
<td>Forecast length</td>
<td>12 hours forecast/analysis cycles</td>
</tr>
<tr>
<td>Assumed data errors</td>
<td>0.8°C (trial and error)</td>
</tr>
<tr>
<td>Ensemble Inflation</td>
<td>5% (trial and error)</td>
</tr>
</tbody>
</table>
| Localization                        | Update single vertical columns
Exponential weight on data errors
(e-folding & cut-off at 100km)      |
| Initial ensemble                    | best initial estimate from model
variability from model run                                                       |

- Same configuration successful in pre-operational tests
Deviation from NOAA Satellite Data

**No assimilation**

- **RMSE of SST forecast (without DA)**
  
  - Map showing deviation with RMS = 1.0577

- **Bias of SST forecast (without DA)**
  
  - Map showing bias with mean error = 0.5298

**Assimilation**

- **over 01.10.2007 - 30.09.2008**
  
  - **ensemble forecast (with LSEIK)**
    - Map showing deviation with RMS = 0.81149

Lars Nerger

Ensemble Data Assimilation with PDAF
Improvement of long forecasts

RMS error over time

black: free model run
Blue/red: 12h assimilation/analysis cycles
green: 5 day forecast

→ Very stable 5-day forecasts
(similar at other dates)
Validation data

- In situ data from MARNET network
- Fixed stations measuring atmosphere and various depths from surface to bottom
- Limited spatial coverage
Validation of forecasts with independent data

- MARNET station data
- Reduction of
  - Bias
  - RMS error

1 year mean over 6 stations:

<table>
<thead>
<tr>
<th></th>
<th>RMSe</th>
<th>bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>free</td>
<td>0.87</td>
<td>0.3</td>
</tr>
<tr>
<td>data</td>
<td>0.59</td>
<td>0.11</td>
</tr>
<tr>
<td>assim.</td>
<td>0.55</td>
<td>0.08</td>
</tr>
</tbody>
</table>

Red: Assimilation 12h forecasts

HBM and ERGOM models

- HBM is operational at BSH and DMI, ERGOM at BSH (currently no data assimilation)
- Model adapted for coastal grids: storage of model fields in vectors of water points (no land mask)
- HBM also used for European Copernicus marine service Baltic Sea (with 4 nested grids; same assimilation framework in testing phase)
- We assimilate into both nested meshes for physics and biogeochemistry
Biogeochemistry: ERGOM model

Atmosphere

\[ \text{PO}_4^{3-} \rightarrow \text{N}_2 \rightarrow \text{O}_2 \]

\[ \text{NO}_3^- \rightarrow \text{N}_2 \rightarrow \text{O}_2 \]

\[ \text{NH}_4^+ \rightarrow \text{N}_2 \rightarrow \text{O}_2 \]

\[ \text{Si} \rightarrow \text{N}_2 \rightarrow \text{O}_2 \]

Ocean

\[ \text{Cyanobacteria} \rightarrow \text{Flagellates} \rightarrow \text{Diatoms} \]

\[ \downarrow \text{Detritus Si} \rightarrow \downarrow \text{Detritus N} \]

Sediment

\[ \text{N}_2 \rightarrow \text{Si} \]

\[ \text{Modified after Maar et al. 2011} \]

www.ergom.net
Grid nesting and data assimilation

State Vector

Physics

Biology

5 km (3 nm) grid

Temperature (°C)

3 km (0.5 nm) grid

Temperature (°C)
Localization in nested grids

Interaction between two different grids at the boundary.

Resolution:
Coarse Grid = 3 nm
Fine Grid = 0.5 nm

Observation location defines influence radius

Used are:
Coarse: 50 km
Fine: 9 km
Assimilation experiments

- Assimilate only SST
- Ensemble size: 20
- March 1 – 31, 2012
- Analysis update every 12 hours
- Filter: LESTKF
- Generate ensemble from model variability over 1 month
- Assimilation experiments
  - weakly coupled: correct only physics; let biogeochemical field react dynamically
  - strongly coupled: correct physics and biogeochemistry
- For strongly coupled DA
  - treat biogeochemistry in log-concentrations (common practice with chlorophyll)
Comparison with assimilated SST data

- Preliminary results
- RMS deviation from SST observations reduced by ~0.2-0.3 °C

Coarse grid:
- little variation over time
- Increasing error-reductions compared to free ensemble run

Fine grid:
- much stronger variability
- partly larger improvement than in coarse grid
- Forecast errors sometimes reach free ensemble run errors
very small changes in weakly-coupled DA case

strong increase of concentration with strongly-coupled DA
Assimilation Influence on Nutrients

Ammonium on March 31, 2012 (micro-mole per m$^{-3}$)

- Very small influence of weakly coupled DA
- Strongly-coupled DA increases concentrations at other locations than Diatoms
Comparison with validation data

- In situ data from DOD and ICES
- Only surface points; 1 month

**Weakly coupled DA**
- Nitrate
- Ammonium

**Strongly coupled DA**
- Nitrate
- Ammonium

**Strong increase of errors in Nitrate and Ammonium**

Nitrate, Ammonium: micro-mole m$^{-3}$
Application Example

Implementation of PDAF for coupled atmosphere-ocean data assimilation
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 (AWI-CM)

Atmosphere
- ECHAM6
- JSBACH land

Ocean
- FESOM
- Includes sea ice

Coupler library
- OASIS3-MCT

Two separate executables for atmosphere and ocean

D. Sidorenko et al., Clim. Dyn. 44 (2015) 757
2 compartment system – strongly coupled DA

might be separate programs

Ensemble Data Assimilation with PDAF
Configure Parallelization – weakly coupled DA

Logical decomposition:
- Communicator for each
  - Coupled model task
- Coupling 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
2 executables ECHAM and FESOM – do all coding twice
  • add subroutine call into both models
  • adapt model communicator (distinct names in the models)
  • replace MPI_COMM_WORLD in communication routines for fluxes

In OASIS-MCT library
  • Replace MPI_COMM_WORLD in OASIS coupler
  • Let each model task write files with interpolation information
Strongly coupled: Parallelization of analysis step

We need innovation: \( d = Hx - y \)

Observation operator links different compartments

1. Compute part of \( d \) on process ‘owning’ the observation
2. Communicate \( d \) to processes for which observation is within localization radius
Execution times (weakly-coupled, DA only into ocean)

MPI-tasks
- ECHAM: 288
- FESOM: 192

Timings (1 day):
- Ens. forecast: 40 – 168 sec
- Analysis step: 0.5 – 0.9 sec

A remaining issue:
- Increasing integration time with growing ensemble size (Factor 4 for 12-fold ensemble size)
- Large variability in integration time over ensemble tasks
- Likely caused by MPI-communication (e.g. no optimal distribution of programs over compute nodes/racks)
Summary

- Unified framework PDAF simplifies implementation and application of data assimilation with existing models
- Application in North & Baltic Seas: Improvement of forecast skill aimed for operational use – assimilation into physical and biogeochemical model components
  - Surface temperature DA successful
  - Strongly coupled DA of temperature deteriorated biogeochemical variables
- Coupled atmosphere-ocean DA with AWI-CM
  - Implementation ready to be used

Thank you!

Lars.Nerger@awi.de  http://pdaft.awi.de
References

- http://pdaf.awi.de