SANGOMA: Stochastic Assimilation for the Next Generation Ocean Model Applications

EU FP7 SPACE-2011-1 project 283580

Deliverable 1.4: Augmented list of common and new tools

Due date: 01/11/2012 Delivery date: 01/11/2012 Delivery type: Report , public

Arnold Heemink Nils van Velzen Martin Verlaan Umer Altaf
Delft University of Technology, NETHERLANDS

Jean-Marie Beckers Alexander Barth
University of Liège, BELGIUM

Peter Jan Van Leeuwen
University of Reading, UK

Lars Nerger
Alfred-Wegener-Institut, GERMANY

Pierre Brasseur Jean-Michel Brankart
CNRS-LEGI, FRANCE

Pierre de Mey
CNRS-LEGOS, FRANCE

Laurent Bertino
NERSC, NORWAY

Table of Contents

1 Purpose of this Document

This documents gives an up to date list of software components that are eligible to be shared in SANGOMA. The modules that are mentioned in this document form a complete set of tools for the initial activities in the SANGOMA project. Later, with more experience gathered in the collaboration we will continue to extend this list.

2 list of common and components

In this section we present up to date list of common components that are suggested by the partners.

By now, the SANGOMA partners have suggested the following modules:

- TUD: POD calibration tool.
- CNRS-LEGOS: ArM, ArM-CA.
- AWI: compute_histogram, compute_stats, eofcovar, mvnormalize, adaptive_forget.
- Ulg: Observation operator for High-Frequency Radar measurements, Weakly Constrained Ensembles (WCE), Gaussian Anamorphosis.
- CNRS-LEGI: geof, lroa, adap, tgop, anam.
- UoR: Sensitivity, Relative Entropy, Mutual Information.

These modules will be described in more detail in the following sections:

2.1 POD calibration tool

- Description of functionality/purpose: Variational method for the calibration of dynamical models. This method does not require an adjoint of the dynamical model. The method uses an approximated adjoint, that is computed only using the forward dynamical model. Proper Orthogonal Decomposition POD or Balanced Proper Orthogonal Decomposition (BPOD) is used for the approximation.
  references:
  - M. U. Altaf, A. W. Heemink, and M. Verlaan, *Inverse shallow-water flow modelling using model reduction*, International Journal for Multiscale Compu- tational Engineering, 7 (2009), pp. 577–596.
  - M. U. Altaf, *Model Reduced Variational Data Assimilation for Shallow Water Flow Models,* PhD Thesis. Delft University of Technology (2011).

- Reqired work: The module is programmed as a result of a PhD project. A partial rewrite is probably needed.
- Inputs: snapshots containing model states.
- Language: F90
- Needs: OpenDA
- Outputs: Optimal solution of the reduces system (parameters)
- Host: TU-Delft

2.2 Observation operator for High-Frequency Radar measurements

- Name of the Module: not decided
- Description of functionality/purpose: this module extracts the radial surface currents from a model state with horizontal scales that are comparable to measurements from a high-frequency radar site.
- Inputs: surface currents
- Outputs: radial currents
- Reqired work: The module needs to be programmed
- Language: ?
- Needs: ?
- Outputs: Optimal solution of the reduces system (parameters)
- Host: ULg/GHER

2.3 Weakly Constrained Ensembles (WCE)

- Purpose: This module creates ensemble perturbations that have to satisfy an a priori linear constraint. It can also be used to create perturbations that are aware of the land-sea mask or that use space- (or time-) dependent correlation length.
- Input: geometry of the domain, correlation length and other fields depending of the nature of the constraints (e.g. velocity field for advection constraint)
- Output: ensemble of constrained perturbations and eigenvalue decomposition of underlying covariance matrix.
- Required work: adaptation of programming interfaces
- Language: Matlab/GNU Octave
- Needs: No additional libraries are required (but NetCDF is recommended)
- Comments: May require data structures larger than 2 GB and may thus require the non-default configure options --enable-64 for GNU Octave.
- Host: ULg

2.4 Gaussian Anamorphosis

- Purpose: Applies a non-linear empirical transformation to a random variables such that the pdf of the transformed variables is closer to a Gaussian distribution
- Inputs: sample of the original variable
- Outputs: transformation function
- Required work: To be seen...
- Language: Matlab/GNU Octave
- Needs: No additional libraries are required
- Comments: No parallelization

- Host: ULg

2.5 ArM

- Description of functionality/purpose :
  - Purpose: Assess the performance of space-time observational arrays at detecting forecast error (as in Le Hénaff, De Mey & Marsaleix, O.Dyn. 59(1), 2009), assess the performance of DA scheme at extracting information from observations
  - Functionality: Representer Matrix spectral analysis, calculation of Array modes, their associated EV spectrum and Modal representers (representers of Array modes in state space)
  - Method: svd(1/sqrt(m-1) * R**(-1/2) * H * Af), where m is the Ensemble size, R is the (not necessarily diagonal) observational error covariance matrix, H is the observation operator for all observations occurring during the current forecast (4D), and Af contains the augmented forecast Ensemble anomalies (4D). The H * Af product is presently implemented here as Bf, the forecast Ensemble anomalies in data space (which has the advantage of being 4D for a modest storage cost, the elements of Bf being collected along the way) – Let us know whether people would favor one or the other formulation.
  - Calling policy: forecast time (end of forecast cycle)
  - Module dependencies: TBD
  - Language: F95
  - Needs: BLAS/LAPACK
- Inputs : 4D forecast Ensemble anomalies in state or data space (TBD), 4D obs. operator (TBD), 4D obs. error cov. matrix or its inverse, square root, or sqrt(inverse)), vector of parameters (number of desired modes, etc.)
- Outputs : eigenspectrum, array modes (data space), modal representers (state space), error code
- Host CNRS-LEGOS

2.6 ArM-CA

- Description of functionality/purpose :
  - Purpose: Array-space consistency analysis
  - Functionality: Project problem of statistical consistency between innovation and Ensemble statistics onto Array-space along Array-mode basis; implement systematic consistency criterion for each rank; provide an overall consistency criterion
  - Method: TBD
  - Module dependencies: TBD
  - Calling policy: forecast time (end of forecast cycle)

    - Language: F95
    - Needs: BLAS/LAPACK
- Module input : TBD, but most likely: 4D forecast Ensemble anomalies in state or data space, 4D obs. operator, 4D obs. error cov. matrix, 4D innovation, array modes and eigenspectrum, vector of parameters (number of desired ranks, etc.)
- Module output : TBD
- Host CNRS-LEGOS

2.7 Compute_histogram

- Description of functionality/purpose : Compute rank histogram of an ensemble about some state (e.g. ensemble mean or true state). Computation is done for a single location. It increments the information stored in a histogram array.
- Inputs: Ensemble values for a single grid point. Single state entry. Size of ensemble. Histogram array (size ensemble size+1. It has to be initialized to zero before the first call).
- Outputs: Histogram array
- Required work: The module needs to be adapted to be generally usable. The input/output is currently different.
- Language: F95
- Needs: no libraries
- Host: AWI

2.8 compute_stats

- Description of functionality/purpose : Compute higher order ensemble statistics (skewness, kurtosis) for a single grid point.
- Inputs: ensemble values for a single grid point, ensemble mean value at the grid point. Size of ensemble.
- Outputs: Values of skewness and kurtosis
- Required work: The module needs to be adapted, because the implementation is not yet generic.
- Language: F95
- Needs: no libraries
- Host: AWI

2.9 eofcovar

- Description of functionality/purpose : Compute EOF decomposition (SVD) of perturbation matrix of some state trajectory. Used to prepare ensemble

generation based on singular vectors/values of state trajectory. It also includes multivariate normalization. The current implementation also includes the reading of the model fields from model-specific files. Also, it writes model-specific outputs. (With PDAF, we always use the same output format for all models. However, the included fields are model-specific.)

- Inputs: Currently, it reads a state trajectory from a Netcdf file.
- Outputs: Currently, it writes the singular vectors, singular values, ensemble mean state as state vectors into a Netcdf file
- Required work: The code can be generalized. SVD computation part can be put into a separate subroutine. Also the file reading and writing can be separated.
- Language: F95
- Needs: NetCDF, LAPACK (DGESVD)
- Host: AWI

2.10 mvnormalize

- Description of functionality/purpose : Normalize an array for unit standard deviation. This is used to perform multivariate normalization in the module 'eofcovar'.
- Inputs: state perturbation array; size of array, offset of a field in the array; size of field
- Outputs: normalized field, value of standard deviation of field before normalization
- Required work: The module is currently part of eofcovar. It could be separated.
- Language: F95
- Needs: no libraries
- Host: AWI

2.11 adaptive_forget

- Description of functionality/purpose : Compute forgetting factor adaptively according to statistical consistency (variance of innovation = 1/forget variance ensemble + variance observations)
- Inputs: ensemble array, ensemble mean, observation vector, size of ensemble array; size of observation vector; ensemble size; default value of forgetting factor
- Outputs: computed forgetting factor
- Required work: The routine is experimental and a core routine of PDAF. It could be generalized for use outside of PDAF.
- Language: F95
- Needs: no libraries

- Comments: The routine is MPI-parallel for domain decomposition. There are two versions for global and local filters.
- Host: AWI

2.12 geof

- Description of functionality/purpose: Compute global EOF decomposition.
- Inputs: Directory with a set of control vectors.
- Outputs: Directory with the EOFs.
- Language: F95
- Needed: in WP4 for CNRS-LEGI.
- Remarks: The algorithm does not require to load the whole ensemble in memory(which is impossible for large size application).
- Host: CNRS-LEGI.

2.13 lroa

- Description of functionality/purpose: Perform square-root observational update, with localization algorithm (as described in Brankart et al., 2011).
- Inputs: Directory with the square root of the background error covariance matrix (as a set of control vectors), the background control vector, the observation vector (following SESAM NetCDF convention), the observation error covariance matrix (assumed diagonal).
- Outputs: Directory with the square root of the updated error covariance matrix, the updated control vector.
- Language: F95
- Needs: NetCDF
- Remarks: The algorithm does not require to load the whole ensemble in memory (which is impossible for large size application), and can be used to update an ensemble forecast.
- Host: CNRS-LEGI.

2.14 adap

- Description of functionality/purpose: Compute optimal inflation factors for the forecast error covariance matrix, and for the observation error covariance matrix (as described in Brankart et al., 2010).
- Inputs: A list of files with statistics describing the previous observational updates (as optionally output by the <lroa> module).
- Outputs: Optimized inflation factors (as optionally input by the <lroa> module).
- Language: F95

- Needs: NetCDF
- Host: CNRS-LEGI.

2.15tgop

- Description of functionality/purpose: Sample truncated Gaussian probability distribution.
- Inputs: Directory with the square root of the (reduced rank) covariance matrix of the reference Gaussian distribution (as a set of control vectors), the mean of the reference Gaussian distribution, directory with the set of linear inequality constraints (as a set of control vectors).
- Outputs: Directory with the required sample (as a set of control vectors).
- Language: F95
- Needs: NetCDF.
- Host: CNRS-LEGI.

2.16 Anam

- Description of functionality/purpose: Estimate and apply anamorphosis transformation (as described in Béal et al., 2011).
- Input : Directory with the input ensemble (as a set of control or observation vectors).
- Outputs: (estimation of the transformation): Directory with a set of quantiles of the input ensemble (as a set of control or observation vectors).
- Outputs: (application of the transformation): Directory with the transformed ensemble (as a set of control vectors).
- Language: F95
- Needs: NetCDF.
- Remarks: The algorithm does not require to load the whole ensemble in memory (which is impossible for large size application).
- Host: CNRS-LEGI.

2.17 Grad_obj

- Description of functionality/purpose: Method for computation of Objective function and gradient. This method is required within in POD calibration module to proceed calibration. Although the aim of this method is to support POD calibration method but can be adapted for any type of variational scheme.
- Inputs:
    - Reduced dynamic operators.

    - Calibration parameters.
- Outputs: Objective function value and gradient vector in reduced space.
- Reqired work: partial implementation required.
- Language: F90
- Needs: OpenDA
- Host: TU-Delft

2.18 Sensitivity

- Description of functionality/purpose: Calculates sensitivity of the posterior mean to the observations (assuming Gaussian observation error and linear observation operator) within a particle filter.
  To be used after weights have been updated by observations.
- Inputs: observation error covariance matrix, (linear) observation matrix, posterior weight vector, particle matrix with individual particle values in each column.
- Outputs: sensitivity matrix of the posterior mean to the observations in observation space, posterior covariance matrix.
- Reqired work: none.
- Language: Matlab
- Needs: no libraries.
- Host: UoR.

2.19 Relative_entropy

- Description of functionality/purpose: Calculates relative entropy (RE) within the particle filter. RE is given by integral{p(x|y)ln[(px|y)/p(x)]}dx. If the particle positions are unchanged during the assimilation, and only the weights are updated then RE can be approximated in terms of the relative weights. Re approx.: Sum(posterior_w/prior_w), where posterior_w and prior_w are the weights of the posterior and prior pdfs, respectively.
  To be used after weights have been updated by observations.
  References:
  Fowler, A. and Van Leeuwen, P. (2012) *Measures of observation impact in non-Gaussian data assimilation.* Tellus A, 64. 17192. ISSN 1600-0870 doi: 10.3402/tellusa.v64i0.17192 .
- Inputs: posterior weight vector, prior weight vector (optional, if not explicitly given these are assumed to be 1/[nr of particles]).
- Outputs: relative entropy, a scalar value.
- Required work: none.
- Language: Matlab.
- Needs: no libraries.
- Host: UoR.

2.20 Mutual_information

- Description of functionality/purpose: Calculates mutual information within the particle filter. Mutual information is the relative entropy averaged over observation space, MI=int(RE*p(y))dy, where p(y) = int[p(y|x)p(x)]dy is given approximately by the sum of the posterior weights assuming likelihood to be Gaussian.
  Mutual information can be approximated in two ways:
  - Quadrature 'quad': discretise the observation space into M (observation space sample size) points, then MI=sum_{i=1}^{M}(RE_i*p(y)_i)*dy.
    Note: only suitable for small observation space.
  - Random sampling 'rndm': sample M (observation space sample size) random points from p(y|x), then MI=sum_{i=1}^M (RE_i*p(y)-i).

  To be used after weights have been updated by observations.
  References:
  Fowler, A. and Van Leeuwen, P. (2012) *Measures of observation impact in non-Gaussian data assimilation.* Tellus A, 64. 17192. ISSN 1600-0870 doi: 10.3402/tellusa.v64i0.17192 .
- Inputs: observation space sample size, vector of spatial the observations at current time, observation error covariance matrix, (linear) observation matrix, particle matrix with individual particle values in each column, string 'quad' or 'rndm' to choose the method of computation, vector of prior weights (optional, if not given these are assumed to be 1/[nr of particle]).
- Outputs: mutual information, a scalar value.
- Required work**:** none.
- Language: Matlab.
- Needs: no libraries.
- Host: UoR.