SANGOMA: Stochastic Assimilation for the Next Generation Ocean Model Applications EU FP7 SPACE-2011-1 project 283580

Deliverable 6.14: Code release 3 documentation

Due date: 31/10/2015

Delivery date: 31/10/2015

Delivery type: Report , public

A. Barth J.-M. Beckers Y. Yan M. Canter
University of Liège, BELGIUM

P.-J. Van Leeuwen S. Vetra-Carvalho
University of Reading, UK

L. Nerger P. Kirchgessner
Alfred-Wegener-Institut, GERMANY

A. Heemink N. van Velzen M. Verlaan U. Altaf
Delft University of Technology, NETHERLANDS

P. Brasseur J.-M. Brankart G. Candille S. Metref
CNRS-LEGI, FRANCE

P. de Mey
CNRS-LEGOS, FRANCE

L. Bertino F. Counillon
NERSC, NORWAY

Executive summary

This deliverable is the third code release of the Sangoma project. Several new tools have been added and adapted to the common data model (deliverable D1.3) . This ensures that the tools can be easily used together and integrated in other assimilation systems. In this release the documentation has also been expanded. Examples have been added on how these tools should be used. The software report of deliverable D2.5 includes the documentation of the tools while this deliverable is the released software bundle with installation instructions.

Intended audience: developers installing the software

Chapter 1

Technical highlights

The following gives an overview of the tools included in this software release. More information about these tools are given in deliverable D2.5. Tools in the colour sangoma blue-green are new tools added in this software release. In total 29 new tools have been developed (often from scratch) and included to this software release which includes now 50 tools in total.

1.1 Diagnostic Tools

Fortran

sangoma_CheckEnsSpread	Compute ensemble spread and deviation of ensemble mean from an input state
sangoma_CheckNormality	Anderson-Darling Test to check normality of a sample
sangoma_CheckWhiteness	Check whiteness of innovations
sangoma_CompareObsDiag	Compare observation-space diagnostics
sangoma_ComputeBRIER	Compute the Brier skill score and its decomposition, and the entropy
sangoma_ComputeCRIGN	Compute CRPF and CRIGN scores
sangoma_ComputeCRPS	Compute the CRPS and its decomposition
sangoma_ComputeEffSample	Compute the effective sample size of a particle filter
sangoma_ComputeEnsStats	Compute ensemble statistics
sangoma_ComputeHistogram	Compute ensemble rank histograms
sangoma_ComputeInvStats	Compute innovation statistics
sangoma_ComputeMutInf	Compute the mutual information
sangoma_ComputeRCRV	Compute the bias & the dispersion of the RCRV
sangoma_ComputeRE	Calculate the relative entropy
sangoma_ComputeSMatrix	Compute scaled ensemble observation anomalies
sangoma_ComputeSensitivity	Calculate the sensitivity matrix with **H** as matrix
sangoma_ComputeSensitivity_op	Calculate the sensitivity matrix with **H** as operator
sangoma_arm	Calculate array modes
sangoma_armca	Check the consistency of an ensemble using array modes
sangoma_ObsDiag	Compute sampled observation-space diagnostics

MATLAB/Octave

computeBRIER	Compute the Brier skill score and its decomposition, and the entropy
computeCRPS	Compute the CRPS and its decomposition
computeRCRV	Compute the bias & the dispersion of the RCRV
computeHistogram	Compute ensemble rank histograms
mutual_information	Compute mutual information in a particle filter
relative_entropy	Compute relative entropy in a particle filter
sensitivity	Compute sensitivity of posterior mean to observations in a particle filter

1.2 Perturbation Tools

Fortran

sangoma_pseudornd2D	Generate random fields with given correlation length
sangoma_MVNormalize	Perform multivariate normalization
sangoma_EOFCovar	Initialize covariance matrix from EOF decomposition

MATLAB/Octave

Weakly constrained ensemble perturbations	Create ensemble perturbations that have to satisfy an a priori linear constraint

1.3 Transformation Tools

Fortran

sangoma_Anamorphosis	Computes local Gaussian anamorphosis
sangoma_ComputeQuantiles	Computes ensemble quantiles as input for anamorphosis

MATLAB/Octave

Empirical Gaussian Anamorphosis	Determine the empirical transformation function such that a transformed variable follows a Gaussian distribution

1.4 Utilities

Fortran

sangoma_computepod	Computes dominant POD modes from an ensemble of snapshots
sangoma_costgrad	Computes the values of Objective function and Gradient using reduced state dimensions
mod_sangoma_utils	Module of utilities for easy porting from MATLAB

MATLAB/Octave

hfradar_extractf	Observation operator for HF radar surface currents

1.5 Analysis

Fortran

sangoma_ens_analysis	Computes the analysis ensemble using the ETKF scheme
sangoma_local_ensemble_analysis	Computes the local analysis ensemble using the ETKF scheme
sangoma_enkf_analysis	Compute analysis ensemble using the EnKF with perturbed observations (globally or with covariance localization)
sangoma_ensrf_analysis	Compute analysis ensemble using the EnSRF with serial observation processing (globally or with covariance localization)
sangoma_estkf_analysis	Compute analysis ensemble using the global ESTKF method
sangoma_etkf_analysis	Compute analysis ensemble using the global ETKF method
sangoma_lestkf_analysis	Compute analysis ensemble using the ESTKF method with observation localization
sangoma_letkf_analysis	Compute analysis ensemble using the ETKF method with observation localization
sangoma_netf_analysis	Compute analysis ensemble using the NETF method

MATLAB/Octave

sangoma_ensemble_analysis	Computes the analysis ensemble using the EnSRF, EAKF, ETKF, ETKF2, SEIK, ESTKF or EnKF scheme
sangoma_local_ensemble_analysis	Computes the local analysis ensemble using the EnSRF, EAKF, ETKF, ETKF2, SEIK, ESTKF or EnKF scheme (domain localization)
sangoma_local_EnKF	Computes the local analysis ensemble using the EnKF (covariance localization)

1.6 Requirements

1.6.1 For Fortran tools

- **GNU make**

- A **Fortran compiler** (such as gfortran, ifort,...). Under Linux, make and gfortran can be installed by the package manager. For Windows and Mac OS, pre-compiled binaries of gfortran are available at http://gcc.gnu.org/wiki/GFortran.

- **LAPACK** and **BLAS**. Reference versions of those libraries are available at http://www.netlib.org/lapack/ and http://www.netlib.org/blas/. An optimized version of BLAS such as OpenBLAS (http://www.openblas.

net/ or Intel's Math Kernel Library (https://software.intel.com/en-us/intel-mkl) should be preferred for better performance.

- **FFTW** for computing the Fast Fourier Transform available at http://www.fftw.org.
- **pkg-config** is used to detect the installation path of FFTW. This package is generally installed by default on a Linux system.

All requirements can be installed on Ubuntu (14.04) by the following command:

```
sudo apt-get install make gfortran libfftw3-dev \
   libblas-dev liblapack-dev pkg-config
```

For Red Hat/Fedora/Centos, these packages can be installed by:

```
yum install make gcc-gfortran fftw-devel blas-devel lapack-devel pkgconfig
```

1.6.2 For GNU Octave/MATLAB tools

Your need to have GNU Octave or MATLAB installed on your machine. GNU Octave is an open-source clone of MATLAB available at http://www.gnu.org/software/octave/.

Under Ubuntu (14.04) you can install octave by the following command:

```
sudo apt-get install octave
```

For Red Hat/Fedora/Centos, octave can be installed by (as root):

```
yum install octave
```

1.7 Installation instructions

Download the file http://sourceforge.net/projects/sangoma/files/sangoma-2.0.tar.gz from the Sourceforge site and decompress it. The decompressed files will be in a folder named sangoma-2.0

```
wget http://sourceforge.net/projects/sangoma/files/sangoma-2.0.tar.gz
tar -xvzf sangoma-2.0.tar.gz
```

1.7.1 Fortran tools

The Fortran tools can be compiled by issuing make in sub-directory Fortran/ of the sangoma-2.0 directory:

```
$ make
gfortran -O3 -fdefault-real-8 -Wall  -I/usr/include  -c sangoma_base.F90 -o sangoma_base.
...
ar: creating libsangoma_tools.a
ranlib  libsangoma_tools.a
```

This creates a library called `libsangoma_tools.a` and some module files. The sangoma tools can be used from a Fortran or C program. A Fortran program must be compiled by using the option `-I` (indicating the path to the module files) and the option `-L` (the path of the library) and the name of the library `-lsangoma_tools`. For example:

```
gfortran -I/path myprogram.f90 -L/path -lsangoma_tools
```

For a C program, only the options -L/path and -lsangoma_tools must be specified.

If a different Fortran compiler is used instead of gfortran, then the variables `FC` and `FFLAGS` in the Makefile need to be adapted. Note that the default floating point type REAL is assumed to be in double precision. A compiler option like `-fdefault-real-8` (gfortran) or `-r8` (ifort) is thus required.

1.7.2 MATLAB/GNU Octave tools

The MATLAB/GNU Octave tools in the sub-directories of the directory MATLAB/ can be used by including the corresponding directory in your MATLAB/GNU Octave search path by using the script `sangoma_addpath` for including all tools. Individual tools can be added by using:

```
addpath(’/path/to/subfolder’);
```

Examples for the tools are provided in the sub-directory examples/ that is include in each category-directory. In case of the Fortran codes, there is a Makefile in the directory that compiles all examples.

1.8 Documentation

Documentation of the tools is available at `http://www.data-assimilation.net/Documents/sangomaDL2.5.pdf`.

1.9 Access to the development version

The code is available in a subversion repository. If you are unfamiliar with subversion you can consult the book Version Control with Subversion, the sourceforge subversion documentation or one of the many tutorials found online (for example Version Tracking With Subversion For Beginners.

1.9.1 Read-only access

You can get the latest version of the sangoma tools by using the following command:

```
svn checkout svn://svn.code.sf.net/p/sangoma/code/tools/trunk sangoma-tools
```

There is no need to be registered for read-only access of the repository. There is also a web-interface available.

1.9.2 Read and write access

Please contact Alexander Barth (a.barth at ulg.ac.be) or Jean-Marie Beckers (jm.beckers at ulg.ac.be) with you sourceforge username (you can register here) and with a small description of the changes that you plan to make. After you have been added as a developer, you can get the development's version by using the following command:

```
svn checkout --username=your_sourceforge_username \
 svn+ssh://your_sourceforge_username@svn.code.sf.net/p/sangoma/code/tools/trunk\
 sangoma-tools
```

You need to replace `your_sourceforge_username` in the previous command.

1.9.3 License

This software release includes code distributed under the terms of the GNU Lesser General Public License (version 3 or later) unless stated otherwise. The source code file contains a reference to the applicable license and the copyright holder.

Deviation from of the Description of Work

This deliverable conforms to the description of work.