

# Use of Ensemble Kalman filters for soil data assimilation

Sam-Erik Walker, William Lahoz  
Norwegian Institute for Air Research - NILU

NetFAM meeting Met.No 17 March 2009

# Motivation

- Build Land DA capability at NILU, including critical mass & capacity
- Match expertise in Met.No & NILU
- Collaboration with world leaders: NWP (Met.No & Meteo-France, HIRLAM); Theory (LMD, France; U. Ghent, Belgium; NERSC, Norway); Land DA (GMU, USA); LSMs (SURFEX, JULES); Observations: Land surface temperature, soil moisture, snow cover (U. Valencia, Spain; SMOS; GlobSNOW)
- Visits to NILU: Paul Houser (Mar 2008); Olivier Talagrand and Pavel Sakov (Feb 2009)
- COST Action ES0804: “Advancing the Integrated Monitoring of trace gas exchange between biosphere and atmosphere”
- ISSI International Team, focus on SMOS soil moisture obs (submit Mar 2009)
- Regular contact between NILU & Met.No (coding; journal club; visits)

# Land DA goals

- (1) Assimilate parameters: land soil temperature (LST), soil moisture, snow (look at, e.g., NH summer 2007 in Europe)
  - (2) Use DA to produce analyses, evaluate observations & models
- 
- (1) Approach:
    - i. Implement EnKF variants into SURFEX
    - ii. Compare with EKF (and other methods) from Météo-France
  - (2) Outcomes:
    - i. Improved use of EO (e.g. for NWP)
    - ii. Forecast capability
    - iii. Improved modelling capability (NWP, climate modelling)
    - iv. Monitoring
    - v. Studies of land/atmosphere systems

# Outlook

- Confront models with observations (extend from NWP)
- Exciting & important problem scientifically with lots of benefits
- Interest to NWP (e.g. HIRLAM)
  - Better use of observations affected by the land
  - Better initial conditions for 2-week to seasonal forecasts
- Interest to climate modellers
  - Better land surface schemes
  - Better climate models
  - Better set up/design of experiments
  - Better understanding of performance of climate models
- General interest to land/atmosphere scientists

# Background: Ensemble Kalman filters

- Since 1994 when Geir Evensen first proposed an ensemble based Kalman filter, there have been many variants of this filter
- An excellent overview of the developments and applications related to this can be found in chapter A.6 in Geir Evensens latest book: “Data Assimilation – The Ensemble Kalman Filter”
- Recently two new “state-of-the-science” versions of the filter have been proposed by Pavel Sakov (now at NERSC) and Peter R. Oke from CSIRO, Australia, which we have implemented at NILU

# EnKF versions at NILU

- Ensemble Square Root Kalman filter (ESRKF) using a symmetric Ensemble Transform Matrix (ETM)
  - Sakov and Oke (2008a): *“Implications of the form of the ensemble transformation in the ensemble square root filters”*, Mon. Wea. Rev., **136**, 1042-1053
- Deterministic Ensemble Kalman Filter (DEnKF) using a linear approximation to the Ensemble Square Root Filter (ESRF) update matrix
  - Sakov and Oke (2008b): *“A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters”*, Tellus, **60a**, 361-371.

# Particle filter versions at NILU

- Importance Sampling and Resampling Particle Filter (SIR)
  - Ristic et al.: *"Beyond the Kalman Filter. Particle filters for tracking applications"*, Artech House, Boston, 2004
- Auxiliary SIR Particle Filter (ASIR)
  - Ristic et al.: *"Beyond the Kalman Filter. Particle filters for tracking applications"*, Artech House, Boston, 2004
- Regularized Particle Filter (RPF)
  - Ristic et al.: *"Beyond the Kalman Filter. Particle filters for tracking applications"*, Artech House, Boston, 2004

These are included in the same Fortran module `ensemble_m.f90` as the two ensemble Kalman filters (and copied into the `enkfassim.f90` code).

# Advantages of EnKF vs EKF (1)

- Some major problems associated with using the EKF in connection with (larger) nonlinear models:
  - Inaccuracy in the evolution of the model error covariance matrix and huge computational requirements associated with the storage and forward integration of this matrix
  - Use of the central forecast as the estimate of the state. For non-linear dynamics the central forecast is not equal to the mean or expected value
- The EnKF was designed to resolve the points above. It has gained in popularity due to its simple conceptual framework and relative ease of implementation
  - No derivation of a tangent linear operator
  - Model error covariance implicitly defined through maintaining a set of model states in the form of an ensemble
  - The mean of the ensemble representing the estimated state



# Advantages of EnKF vs EKF (2)

- In EnKFs (and particle filters) each ensemble member is run forward in time through the model
- Uncertainty (or spread in the ensemble) is introduced by stochastic model dynamics (stochastic physics) when integrating each ensemble member forward in time
- In the EKF uncertainty in the estimated state is introduced in the update of the B-matrix (background error covariance) and in the added Q-matrix
- However, both are optimal and correct strictly speaking only when the underlying PDFs (prior and posterior to the observations) are Gaussian

# Advantages of particle filters

- A particle filter can handle PDFs which can be quite different from Gaussian, e.g. skewed, heavy-tailed, bi-modal, multi-modal etc.
- However, in order to take advantage of this often a *very large* ensemble is needed
- But another nice feature of particle filters is that they do not alter model physics, only reassigns the probabilities of model states
- In all ensemble based methods, EnKFs, particle filters, the ensemble should capture the true state well, ideally through the mean, and should also describe well the uncertainty of the true state via the spread of the ensemble

# Ensemble Square-Root Kalman Filter (1)

Sakov & Oke 2008a

$$\mathbf{x}^f = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i^f$$

$$\mathbf{A}^f = \left[ \mathbf{A}_1^f, \dots, \mathbf{A}_N^f \right] = \left[ \mathbf{X}_1^f - \mathbf{x}^f, \dots, \mathbf{X}_N^f - \mathbf{x}^f \right]$$

$$\mathbf{P}^f = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{X}_i^f - \mathbf{x}^f)(\mathbf{X}_i^f - \mathbf{x}^f)^T = \frac{1}{N-1} \mathbf{A}^f \mathbf{A}^{fT}$$

# Ensemble Square-Root Kalman Filter (2)

Sakov & Oke 2008a

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^f)$$

$$\mathbf{K} = \mathbf{P}^f \mathbf{H}^T (\mathbf{H}\mathbf{P}^f \mathbf{H}^T + \mathbf{R})^{-1}$$

Kalman filter  
equations

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f$$

# Ensemble Square-Root Kalman Filter (3)

## Sakov & Oke 2008a

$$\mathbf{A}^a = \mathbf{A}^f \mathbf{T}$$

$$\mathbf{T} = \left[ \mathbf{I} + \frac{1}{N-1} (\mathbf{H}\mathbf{A}^f)^T \mathbf{R}^{-1} (\mathbf{H}\mathbf{A}^f) \right]^{-1/2}$$

$$\mathbf{I} + \frac{1}{N-1} (\mathbf{H}\mathbf{A}^f)^T \mathbf{R}^{-1} (\mathbf{H}\mathbf{A}^f) = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

$$\mathbf{T} = \mathbf{V}\mathbf{\Lambda}^{-1/2}\mathbf{V}^T$$

Singular value decomposition with  $\mathbf{V}$  orthonormal and  $\mathbf{\Lambda}$  diagonal with eigenvalues

# Ensemble Square-Root Kalman Filter (4)

## Sakov & Oke 2008a

$$\mathbf{X}^a = [\mathbf{X}_1^a, \dots, \mathbf{X}_N^a] = [\mathbf{A}_1^a + \mathbf{x}^a, \dots, \mathbf{A}_N^a + \mathbf{x}^a]$$

$$\sum_{i=1}^N \mathbf{A}_i^a = \mathbf{0} \quad \text{Mean preserving T} \quad \rightarrow \quad \mathbf{x}^a = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i^a$$

$$\mathbf{P}^a = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{X}_i^a - \mathbf{x}^a)(\mathbf{X}_i^a - \mathbf{x}^a)^T = \frac{1}{N-1} \mathbf{A}^a \mathbf{A}^{aT}$$

Kalman filter equations exactly solved for the ensemble mean

Note: No perturbations of observations

# Ensemble Square Root Kalman Filter (5)

## Sakov & Oke 2008a – NILU subroutine

```
call ensrkf(ndim, nens, nobs, Xf_ens, xf,  
            Yf_ens, yf, y, R, Xa_ens, xa)
```

ndim = Number of state variables

nens = Number of ensemble members

nobs = Number of observations

Xf\_ens = Forecasted ensemble ndim x nens

xf = Mean of forecasted ensemble ndim

Yf\_ens = Forecasted (simulated) observations nobs x nens

yf = Mean of forecasted (simulated) observations nobs

y = Real observations nobs

R = Diagonal of R matrix (observation errors) nobs

Xa\_ens = Analysed ensemble ndim x nens

xa = Mean of analysed ensemble (the analysed state) ndim

# Implementation issues (1)

- Current code at NILU based on Fortran 90, BLAS (Basic Linear Algebra Subprograms), Sparse BLAS and LAPACK
- Observation matrix free version implemented using only observation operator function  $h(x)$ , with no reference to the observation matrix  $H$ 
  - Ref.: Mandel J., Efficient implementation of the Ensemble Kalman filter, UCDHSC/CCM report no. 231, Univ. of Colorado, May 2006.
- Current implementation has computational complexity which is linear in  $n$  (# var) and cubic in  $p$  (# obs)



# Implementation issues (2)

- If  $p \gg n$  other more efficient implementations of the EnKF can be implemented which are linear in both  $n$  and  $p$ 
  - Ref.: Mandel J., Efficient implementation of the Ensemble Kalman filter, UCDHSC/CCM report no. 231, Univ. of Colorado, May 2006.
- These implementations are possible when the obs. covariance matrix  $R$  is diagonal or banded, and easy to decompose
- A numerically stable square-root version involving symmetric products of matrices and Cholesky decomposition seems promising
  - See Ch. 4.3 in the above report

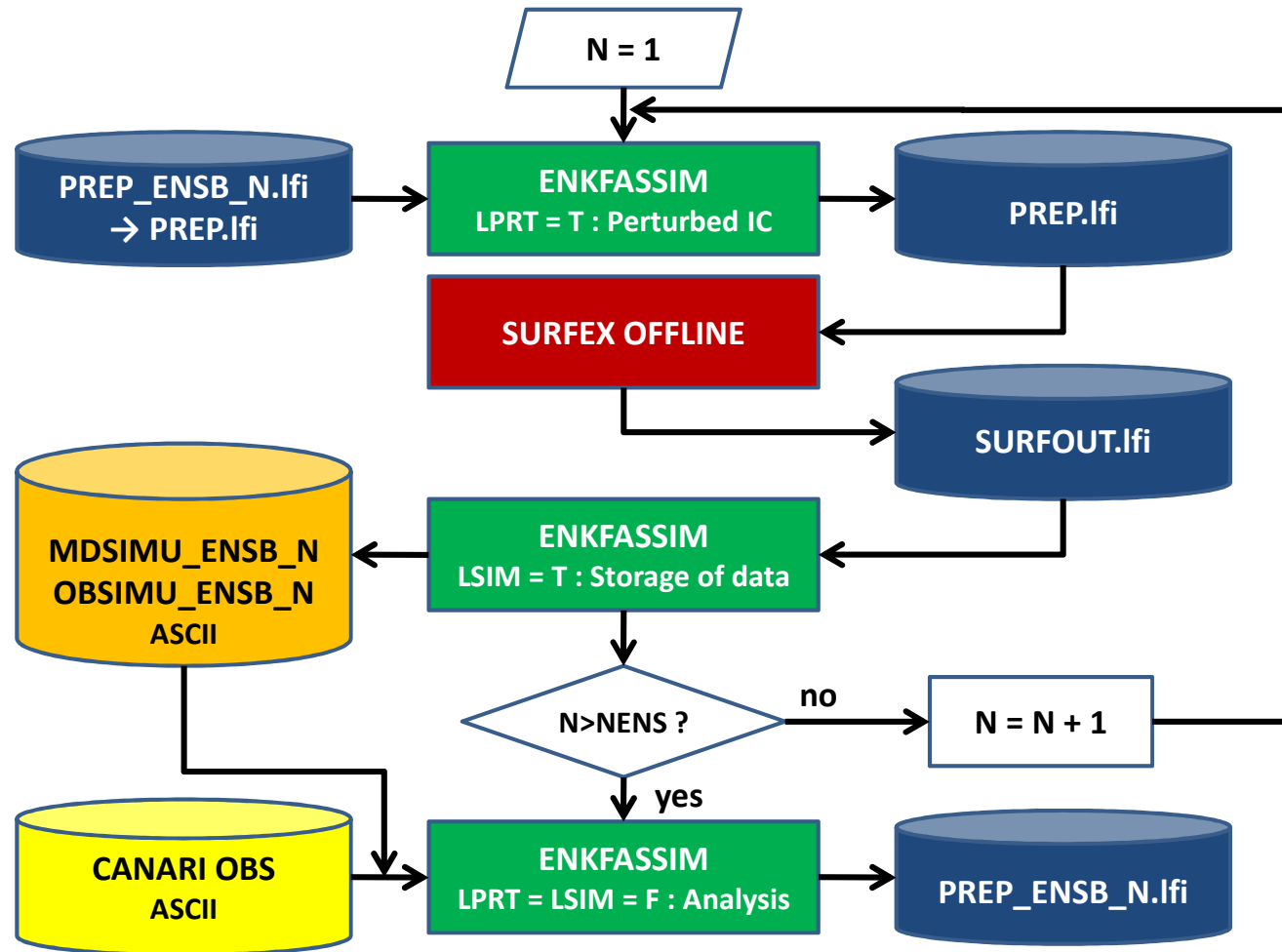
# SURFEX EnKF

- Based on the offline version of SURFEX v4.7 and the current SURFEX-EKF LDAS code, script and documentation kindly provided by Jean-François Mahfouf
- Currently installed in a Linux based computer system at Met.No facilities in Oslo
- Thanks to Viel Ødegård, Mariken Homleid and Trygve Aspelien at Met.No for all help in installing and running the SURFEX EKF from which the SURFEX EnKF is now being built!

# SURFEX EnKF

- New run script: run\_enkf.sh
- New main program: enkfassim.f90
- New makefile: Makefile.SURFEX.mk
- New namelist file: OPTIONS.nam
  
- Otherwise identical to the SURFEX-EKF system

# Flowchart of SURFEX-EnKF



# SURFEX EnKF: OPTIONS.nam

```
&NAM_IO_ENKFASSIM  
LPRT = F,  
LPRF = F,  
LSIM = F  
/
```

LPRT associated with perturbation (random draw) of each control variable in order to create new ensemble members.

LPRF associated with perturbation of forcing data in order to create new ensemble members. Currently this option is not used.

LSIM is used to read and write model variables and simulated observations as in the SURFEX-EKF version.

# SURFEX EnKF: OPTIONS.nam

```
&NAM_VAR
IVAR          = 1,
NVAR          = 1,
XVAR_M(1)    = 'WG2',
XVAR_M(2)    = 'WG1',
XVAR_M(3)    = 'TG2',
XVAR_M(4)    = 'TG1',
...
TPRT_M(1)    = 0.1,
TPRT_M(2)    = 0.1,
TPRT_M(3)    = 0.1,
TPRT_M(4)    = 0.1,
INCV(1)      = 1,
INCV(2)      = 1,
INCV(3)      = 0,
INCV(4)      = 0
/
```

Currently the EnKF runs with the two-layer version of the ISBA scheme. It means that the control variables can be the four main prognostic variables of this scheme:

The surface temperature  $T_s$  (TG1), the mean surface temperature  $T_2$  (TG2), the superficial volumetric water content  $w_g$  (WG1), and the mean volumetric water content in the root-zone  $w_2$  (WG2).

TPRT is the standard deviation of Gaussian random variables for perturbation of the logarithm of each control variable (mean values assumed to be zero).

# SURFEX EnKF: OPTIONS.nam

/

&NAM\_OBS

NOBSTYPE = 3,

YERROBS(1) = 1.0,

YERROBS(2) = 0.1,

YERROBS(3) = 0.4,

INCO(1) = 1,

INCO(2) = 1,

INCO(3) = 1

/

Regarding the observations, three observation types are considered:

Screen level temperature, relative humidity and superficial soil moisture content.

Like for the control variables, the elements of the array INCO control which type of observation one wants to assimilate.

# SURFEX EnKF: OPTIONS.nam

&NAM\_ENKF

IENS = 1,

NENS = 1,

ENKFM = 1

/

IENS is currently not used. It may be used in the future to operate on a single ensemble member

NENS denotes the number of ensemble members

ENKFM denotes the type of ensemble method:

1: Ensemble square-root Kalman filter: ENSRKF

2: Deterministic ensemble Kalman filter: DENSKF

3: Importance sampling and resampling particle filter: SIR

4: Regularized particle filter: RPF

All ensemble data assimilation methods are based on BLAS, Sparse BLAS and LAPACK subroutine libraries. Generation of pseudo random numbers based on subroutines using the Mersenne Twister RNG.



# Possible issues for discussion (1)

- How to best generate an initial ensemble?
- How to perturb ensemble members as we move along?  
Introduction of stochastic physics errors in SURFEX and/or in the forcing data?

The current control variables are:

- Surface temperature  $T_s$  (TG1)
  - Mean surface temperature  $T_2$  (TG2)
  - Superficial volumetric water content  $w_g$  (WG1)
  - Mean volumetric water content in the root-zone  $w_2$  (WG2)
- How many ensemble members are needed?

# Possible issues for discussion (2)

- How to perform bias correction?
- How to optimally tune observation errors?
  - Application of Desroziers method?
- How to introduce localized variants in connection with a possible 3D version of SURFEX EnKF?
- Other variables of interest to assimilate: Snow cover,...