



*Supplement of*

**Development and evaluation of 0.05° terrestrial water storage estimates using Community Atmosphere Biosphere Land Exchange (CABLE) land surface model and assimilation of GRACE data**

**Natthachet Tangdamrongsub et al.**

*Correspondence to:* Natthachet Tangdamrongsub ([natthachet.tangdamrongsub@nasa.gov](mailto:natthachet.tangdamrongsub@nasa.gov))

The copyright of individual parts of the supplement might differ from the article licence.

## Supplement

### **Pseudocodes of GRACE DA**

CABLE LSM is written in Fortran90, while the GRACE DA module can be written in any programming language, e.g., Fortran, Python, Matlab. In this study, the processing is performed in a Unix environment with Fortran compiler, and NetCDF library installed. Comprehensive details on CABLE installation and simulation are given in <https://trac.nci.org.au/trac/cable/wiki/CableUserGuide>. A single-core central processing unit (CPU) and 16 GB memory are sufficient to carry out the simulation, but higher computational power helps to speed up the process. The processing details of GRACE DA are clearly described in Sect. 3 and Fig. 3. Below is the pseudocode of the processing.

#### **S1. Main program**

Define  $Nlat$  = Length of latitude of the study domain,  $Nlon$  = Length of longitude

Define  $Ntime$  = Number of simulation epoch,  $Nstate$  = number of state variables,  $Nens$  = number of ensemble members

Set first processing year and month

**While** processing year  $\neq$  last year and month  $\neq$  last month

    Perturb forcing data and parameter of the processing year and month (see S2 below)

    Perform model propagation of all ensemble members for one month

**If** GRACE data of the processing year and month are available

        Print “Performing GRACE DA”

        Perturb GRACE data of the processing year and month (see S2 below)

        Collect ensemble model TWS states in matrix **A** (see Sect. 3.1)

        Collect perturbed TWS observation in matrix **D** (see Sect. 3.1)

        Define matrix **H** (see Eq. 3 and S3 below)

        Compute TWS monthly update ( $\Delta\mathbf{A}$ ) using Kalman equation (Eq. 4)

        Divide TWS monthly update by a number of days ( $\Delta\mathbf{A}/N_{\text{day}}$ ) to obtain the daily update

        Reinitialize initial states

**For** ensemble member  $j$

**For** process day  $d$

                Add TWS daily update  $j$  to TWS initial states of day  $d$

                Perturb forcing and parameter of year, month, and day  $d$  (see S2 below)

                Perform model propagation of ensemble  $j$  for day  $d$

Save simulation output of ensemble  $j$  for day  $d$  in memory

**End**

**End**

**Else**

Print “Missing observations, skip GRACE DA”

Do nothing

**End**

Collect result, e.g., compute ensemble mean of states (see S4 below)

Set processing year and month to the next processing year and month

**End**

---

## **S2. Perturbation**

Obtain forcing data, parameter, or observation from input

Obtain the uncertainty (standard deviation,  $std$ ) of forcing data, parameter, or observation from input

Obtain correlation length ( $L$ ) of forcing data and observation from input

**If** the input is forcing data

Define matrix **F**; dimension is 4D ( $Nlat, Nlon, Ntime, Nens$ )

**If** the input is precipitation or shortwave radiation

Generate multiplicative noise based on lognormal distribution with given  $std, Nens, L$

**Else if** the input is temperature

Generate additive noise based on a normal distribution with given  $std, Nens, L$

**End**

**For** each ensemble member  $j$

Apply the generated noise to nominal value and store in **F**[ : , : , : ,  $j$  ]

**End**

**Else if** the input is parameter or observation

Define matrix **P**; dimension is 3D ( $Nlat, Nlon, Nens$ )

Generate additive noise based on a normal distribution with given  $std$  and  $Nens$

**For** each ensemble member  $j$

Apply the generated noise to the nominal value and store it in **P**[ : , : ,  $j$  ]

**End**

**Else if** the input is observation

Define matrix **OBS**; dimension is 3D ( $Nlat, Nlon, Nens$ ) to store perturbed monthly TWS

Generate additive noise based on a normal distribution with given  $std, Nens, L$  ( $L$  is optional)

**For** each ensemble member  $j$

Apply the generated noise to the nominal value and store it in **OBS**[ : , : ,  $j$  ]

**End**

**End**

---

### **S3. Setting matrix H**

Obtain model and mascon coordinate (latitude, longitude) from input

Determine **H** matrix dimension based on the number of model grid cells inside mascon cells

Define zero matrix **H**

**For** mascon grid cell  $i$

Search for all model grid cells ( $k$ ) inside mascon  $i$ , and create a row vector

Assign  $1/k$  element to the row vector

Place the row vector at  $(i, k_i)$  position of the matrix **H** (Eq. 3)

**End**

---

### **S4. Saving ensemble results**

Each ensemble output can be obtained from the memory

Allocate matrix **B**; dimension is 5D ( $Nlat, Nlon, Ntime, Nstate, Nens$ ) to store ensemble outputs

**For** ensemble output  $j$

Collect daily estimates of all estimated variables in matrix **B** [ : , : , : , : ,  $j$  ]

**End**

Compute ensemble mean value of the matrix **B**; the output is a 4D matrix ( $Nlat, Nlon, Ntime, Nstate$ )

Compute ensemble standard deviation value of the matrix **B**; the output is a 4D matrix

Save monthly ensemble mean and standard deviation in the desired file format (e.g., NetCDF)

---