ARPSEnKF

Mingjing Tong[1]

1. Source files

List of ARPS files modified for EnKF:

Makefile
makearps
src/arps/initpara3d.f90
src/arps/micro_lfo_ice.f90 (modified for parameter estimation)
src/arps/outlib3d.f90

src/include/globcst.inc
src/include/grid.inc
src/include/phycst.inc

src/arpsintrp/Makefile
src/arpsens/Makefile

ARPS EnKF files:

Initialize ensemble members:

src/arpsens/arpsenkfic.f90
src/arpsens/initenkfic.f90
src/arpsens/rndprt.f90
input/arpsenkfic.input
src/arpsintrp/radardtaintrp.f90
input/radardtainput.input

EnKF analysis code:

src/arpsenkf/README.enkf
src/arpsenkf/Makefile
src/arpsenkf/.f90*
include/enkf.inc
include/arpsenkfic.inc
input/arpsenkf.input

Scripts and examples:

*scripts/EnKF/**

[1] Original author: Dr. Tong. The original code was within ARPS version 5.2.6 from Dr. Tong on 10/31/2007. It was merged with the latest ARPS version (arps5.2.9) by Yunheng Wang and will be managed officially. This document was also modified slightly by Yunheng Wang on 01/02/2008.

2. Script files

Inside *$arpsrootdir/scripts/EnKF/scripts/machine* resides script file *runenkf*, which is the main script for running the forecast and assimilation cycles. Here *machine* is a machine name such as *rachel* and *cyclone*, for which the script is specifically configured for.

arpsrootdi*r*: the ARPS root directory,
e.g., `arpsrootdir = /home/mtong/arps5.2.6Beta`
Inside script file *runenkf*, the following variables need to be defined:
Make sure you specify the full directory path.

casedir: The directory where you want to run your job
Directory $*casedir* should contain namelist input files

arps.input	input file for ARPS forecasts. A default or template is available inside $*arpsrootdir/input*.
arpsenkfic.input	input file for **arpsenkfic**. A default or template is available inside $*arpsrootdir*/*input*.
arpsenkf.input	input file for **arpsenkf**. A default or template is available inside $*arpsrootdir*/*input*.

and script files

runenkf main script file for running the forecast and assimilation cycles.

- Before you run ARPS EnKF, make sure that the parameters in the input files and scripts have been set correctly.

expnam: The name of your experiment
In input file *enkf.input*, the following namelist parameter should be set to $*expnam*:
casenam = '$expnam',
In input file **arps.input**, the following namelist parameters should be set to $*expnam*:

```
runname = '$expnam',
inifile = '$expnam.hdfenfo',
inigbf  = '$expnam.hdfgrdbas',
```

Here, $*expnam* means the value of variable *expnam* set in script *runenkf*. Do not directly use character string $*expnam* inside the namelist input files.

Before running the job, copy $*arpsrootdir*/*input*/*arps.input*, $*arpsrootdir*/*input*/*arpsenkfic.input and* $*arpsrootdir*/*input*/*arpsenkf.input* into *$casedir***.** And directory called $*arpsrootdir*/*EnKF*/*example* already exists in the example package and certain parameters have been set for experiment ‘may20’. If you want to try this experiment, do not recopy *arps.input* or *enkf.input* - otherwise, you will overwrite them using the default *arps.input* and *arpsenkf.input* files.

workdir: The name of the directory in which forecast and assimilation cycles run and your EnKF data output are saved.

- Typically a scratch disk is used for this purpose.
- Do not use the same name for *expnam* and *datadmpdir*, since script *runenkf* will edit *arps.input* and change the value of *expnam* to *enfxxx*.

resultdir: The name of the directory into which useful EnKF data output are saved. (optional)

radardatadir: The directory containing input radar data for **arpsenkf**. For OSSE, the data will be generated by program **ossedata** and written into this directory.

- The data could be *simulated* data or *real* data. Simulated data need to be first generated by ***ossedata*** before running ***arpsenkf***.

truthsimdir: The directory containing truth simulation data.

executdir: directory containing executable files (*arps* or *arps_mpi*, *arpsenkfic*, *arpsenkf*, *ossedata*, etc.)

- By default, the executables are placed inside $*arpsrootdir*/*bin*, so *executdir* = $*arpsrootdir*/*bin*. If you are running multiple experiments simultaneously that require different executables (because of code modifications), you can move your executables into a subdirectory, e.g., $*arpsrootdir*/*bin*/*osse*.

nn: ensemble size or the number of ensemble members

nn = nen set inside enkf.input
PBS_VPPN: The number of processors required for ensemble forecasts and EnKF analysis that are run simultaneously at a time.

- $*nn* ensemble members will be divided into $*nn*/$ *PBS_VPPN* groups. In each group, the model integration of $ *PBS_VPPN* members will be run simultaneously. The groups will be run until all ensemble forecasts are completed, group by group.

group: The number of groups for initialize ensemble member or for ensemble forecast

it is equal to $*nn*/$ *PBS_VPPN*
tsta: The model time (seconds) of your initial ensemble forecasts

tinter0: The forecast length (seconds) of the first set of ensemble forecasts starting from $*tsta* *tinter0* can be the same as or different from *tinter* defined below

tinter: The time interval (seconds) between successive EnKF analyses, or the length of assimilation cycles (currently the assimilation cycles are of the same length, at least within the same job)

- in file *arps.input*, the following namelist parameters need to be set to be consistent with the above parameters, as follows:

```
&timestep
   tstart =   $tsta,
   tstop  =   $tsta + $tinter0,

&history_dump
   thisdmp  = $tinter0,
   tstrtdmp = $tsta,
```

cycle: The number of assimilation cycles

- It should be set so that $*tsta* + $*tinter0* + $*cycle* × $*tinter* is the last analysis time or the end of the assimilation window.

3. Input files

A template namelist input file for ARPS ensemble forecasts, *arps.input*, is needed that is edited by *runenkf* script to produce actual input files used by individual ensemble members.

Comparing the ARPS input files used by the truth simulation of OSSE and by the ensemble forecasts during EnKF cycles, a number of parameters, including those listed in the following, are different.

```
&initialization
   pt0opt = 0,
```

All parameters need to be set properly for the truth simulation. See, e.g., *arps.input* in $arpsrootdir/simulation/may20 for the May 20 supercell case.

```
&timestep
   tstart =   $tsta,
   tstop  =   $tinter0,

&history_dump
   thisdmp  = $tinter0,
   tstrtdmp = $tsta,
```

For the ensemble forecasts, tsta and tinter0 are set to their values specified in runenkf. For the truth simulation, they should be according the simulation start time (*tstart*) and stop time (*tstop*) and the desired history dump interval (*thisdmp*) and start time (*tstrtdmp*).

Namelist input file *arpsenkf.input* contains all the namelist parameters controlling the EnKF analysis. Most of the parameter settings are explained inside *arpsenkf.input*. More information the EnKF system can be found in *src/arpsenkf/README.enkf*.

4. Output files

1) Output of **ossedata**, a program that creates simulated radar data from ARPS history dumps

$wsrnam_$expnam_tttttt or $casanam_$expnam_tttttt in directory $rdrobsdir

- '$' plus parameter name means the values set in script *runenkf* or in *arpsenkf.input*
- For example, $*wsrnam*_$*expnam_tttttt* could be *KTLX_may20_2km_001500*
- '*tttttt*' is the model time in seconds.

$trusimnam.hdftrureftttttt

- The truth reflectivity data at the scalar points of model grid calculated from the truth simulation fields
- Used for verification in *rmserror.f90*

2) Output of *arpsenkf*, the EnKF analysis program
efmean.hdftttttt: ensemble mean forecast
enmean.hdftttttt: ensemble mean analysis
enfxxx.hdftttttt: input for *arpsenkf* (output of ensemble forecast) by ensemble member *xxx*. In *runenkf*, it is renamed as *enfxxx.hdfenkfi*
enaxxx.hdftttttt: output of *arpsenkf* (input of ensemble forecast) by ensemble member *xxx*. In *runenkf*, it is renamed as *endxxx.hdfenkfo*.

3) Diagnostic output data for OSSE:
emftttttt: rms error of ensemble mean forecast
ematttttt: rms error of ensemble mean analysis
esftttttt: prior or forecast ensemble spread
esf_bfinftttttt: prior or forecast ensemble spread before covariance inflation or adding assumed model error
esattttttt: posterior or analysis ensemble spread

The following data are dumped when *rmsesprddmp* = 1 in arpsenkf.input
emf1Dtttttt.dat: vertical profiles of the horizontal average rms errors of ensemble mean forecast

ema1Dtttttt.dat: vertical profiles of the horizontal average rms errors of ensemble mean analysis
esf1Dtttttt.dat: vertical profiles of the horizontal average forecast ensemble spread
esf1D_bfinftttttt.dat: vertical profiles of the horizontal average forecast ensemble spread before covariance inflation or adding assumed model error.
esa1Dtttttt.dat: vertical profiles of the horizontal average analysis ensemble spread

${*radarname*}*bgtttttt*: ensemble mean forecast projected to observation space
${*radarname*}*antttttt*: ensemble mean analysis projected to observation space
${*radarname*}*innovtttttt*: Innovation statistics verified against radar with name ${*radarname*}

${*radarname*}*rmsdbgtttttt*: rms difference between ensemble mean forecast and radar observation.
${*radarname*}*rmsdantttttt:* rms difference between ensemble mean analysis and radar observation.

${*radarname*}*rmsdsndtttttt:* rms difference between storm environment and radar observation.

${*radarname*}*spreadanatttttt*: analysis ensemble spread in radar observation space

${*radarname*} can be $*wsrname* or $*casaname*

The following are output of parameter estimation
paraem_b_tttttt: ensemble mean of estimated parameter before analysis
paraem_a_tttttt: ensemble mean of estimated parameters after analysis
parans_a_tttttt: 1σ ensemble width of estimated parameter (negative width)
parans_a_tttttt: 1σ ensemble width of estimated parameter (positive width)

rmscorttttttt: rms correlation coefficient between estimated parameter and radar observations Vr and Z.

corvrtttttt.dat: 3D correlation coefficient between estimated parameter and Vr (the data can be plotted using ppiplt.f90)
corvrtttttt.dat: 3D correlation coefficient between estimated parameter and Z (the data can be plotted using ppiplt.f90)

5. Running arpsenkf

Compilation

```
cd arps5.2.6Beta
makearps clean
```

makearps *arps* or **makearps** *arps_mpi*
makearps *arpsenkfic*
makearps *rndprt* (only if run parameter estimation experiments or perturbing initial environment state)

makearps *–omp arpsenkf* (*-omp* option for machines that support OpenMP only)
makearps *ossedata* (for OSSE only)
makearps *endgns* (ensemble diagnostic program)

Make directory to run you job (called job directory)

1. Choose the right version of *runenkf* script from $*arpsrootdir /EnKF/scripts/machine* and copy it into your job directory

Make work directory (and results directory) as defined by $*workdir* (and $*resultdir*) in *runenkf*
Copy default *arps.input* and *arpsenkf.input* from $*arpsrootdir/input* to your job directory $*casedir*
Set parameters in script and input files (see section 2-3)
Run *runenkf* script inside a batch script or interactively.

NOTE: For an OSSE, before running *arpsenkf* , you need to first create the simulated data first, which involves a truth simulation and simulated data creation. The steps are as follows. These steps are needed only once.

a) Perform the truth simulation. For example:

(1) Make new directory for your truth simulation
(2) Copy *arps.input* to $*trusimdir* and set parameters in *arps.input*
(3) Run **arps** to generate the simulation data, in the form of ARPS history dumps.

In *arps.input*, *dirname* = '$*trusimdir'* sets the directory where the truth simulation output are saved.

b) In your job directory, set parameters in *enkf.input* (see *README.enkf* in *src/arpsenkf/*) and run **ossedata** to generate simulated radar radial velocity and reflectivity data.

In *enkf.input*, *obsdatadir* = '$ *rdrobsdir*' specified the directory into which the simulated radar data are saved.

2. An example

For the May 20 Del City case, 2 km resolution without vertical stretching.
The example is in $*arpsrootdir/EnKF/example/may20_2km*

An example for generating truth simulation is in $*arpsrootdir/ EnKF/example/simulation*

An example for generating simulated radial velocity and reflectivity data is in $*arpsrootdir/ EnKF/example/ossedata*

6. Other support and plot programs

arps5.2.3_EnKF/EnKF/helpsrc/*readinnov.f90*: read innovation statistics at different time into one data file.

arps5.2.3_EnKF/EnKF/helpsrc/*paraestdata.f90*: read mean and 1σ ensemble width of estimated parameters at different time into one data file.

The output of *readinnov.f90* and *paraestdata.f90* can be directly plotted using **Grace**.

arps5.2.3_EnKF/EnKF/pltsrc/*rmseplt.f90*: plot rms errors and ensemble spread.
arps5.2.3_EnKF/EnKF/pltsrc/*rmsprofile.f90*: plot vertical profile of rms errors and spread.

arps5.2.3_EnKF/EnKF/pltsrc/*ppiplt.f90*: plot radar data (both simulated and real data on radar elevation levels and on model scalar columns)