Acknowledgement

We thank the National Oceanic and Atmospheric Administration (NOAA) Office of Oceanic and Atmospheric Research (OAR) for their support of this work. This work is also facilitated by the National Center for Atmospheric Research (NCAR). NCAR is supported by the National Science Foundation (NSF).
Foreword

This User’s Guide for the community ensemble Kalman filter (EnKF) data analysis system is particularly geared for beginners. It describes the fundamentals of using EnKF, including basic skills of installing, running, diagnosing, and tuning EnKF. EnKF version (v) 1.3 was released in November 2018. This version of code is compatible with the Gridpoint Statistical Interpolation (GSI) analysis system community release v3.7.

This User’s Guide includes six chapters and one appendix:

**Chapter 1** provides a background introduction of the operational and community EnKF system, Data Assimilation review committee, and data types that can be used in this version.

**Chapter 2** contains basic information about how to get started with EnKF, including system requirements; required software; how to download EnKF; and information about compilers, libraries, and how to build the code.

**Chapter 3** focuses on the input files needed to run EnKF and how to configure and run GSI observer and EnKF through a sample run script. This chapter also provides an example of a successful EnKF run.

**Chapter 4** includes information about diagnostics and tuning of the EnKF system through EnKF standard output and namelist variables.

**Chapter 5** illustrates how to setup and run the GSI observer and EnKF for a regional configuration and a global configuration, as well as how to diagnose the results.

**Chapter 6** introduces EnKF theory and the main structure of the code.

**Appendix A** describes the contents of the EnKF namelist.

This document is updated annually. For the EnKF documentation and annual release code, please visit the EnKF User’s Website:

https://dtcenter.org/EnKF/users

Please send questions and comments to the EnKF help desk:

enkf-help@ucar.edu

This document and the annual EnKF releases are made available through a community EnKF effort led by the Developmental Testbed Center (DTC), in collaboration with EnKF developers. To help sustain this effort, we encourage for those who use the community released EnKF, the EnKF helpdesk, the EnKF User’s Guide, and the other DTC EnKF services, please refer to this user’s guide in their work and publications.
# Contents

## 1. Overview

1.1. EnKF History and Background ............................................. 1
1.2. EnKF Becomes Community Code ........................................... 2
   1.2.1. EnKF Code Management and Review Committee .................. 2
   1.2.2. Community Code Contributions ..................................... 2
1.3. About This EnKF Release .................................................. 3
   1.3.1. What Is New in This Release Version ............................... 3
   1.3.2. Observations Used by This Version ................................. 4

## 2. Software Installation

2.1. Introduction ............................................................... 6
2.2. Obtaining the Source Code .............................................. 7
   2.2.1. Versions of EnKF .................................................... 7
2.3. Compiling EnKF ............................................................ 8
   2.3.1. Example build using the Intel Fortran Compiler ................. 9
   2.3.2. Default Options .................................................... 10
2.4. System Requirements and Compiler Specific Notes .................... 10
2.5. Compilers Tested for Release ........................................... 11
2.6. Getting Help and Reporting Problems .................................. 11

## 3. Running EnKF

3.1. Input Data Required to Run EnKF ....................................... 12
   3.1.1. Input Data Required to Run EnKF ................................. 12
3.2. EnKF and GSI Observer Run Scripts ................................... 14
   3.2.1. General Introduction to the Run Scripts ......................... 14
   3.2.2. GSI Observer Run Scripts ......................................... 15
   3.2.3. Sample Regional EnKF Run Scripts ................................. 19
3.3. Understanding Resulting Files in GSI Observer and EnKF Run Directory ......................................................................................................................... 23
   3.3.1. The GSI Observer Run Directory .................................... 23
   3.3.2. Files in the EnKF Run Directory .................................... 24

## 4. EnKF Diagnostics and Tuning

4.1. Understanding the Standard Output from EnKF (stdout) .............. 26
Contents

4.2. Tuning of Inflation and Localization .................................................. 33
4.3. Tuning EnKF through Key Namelist Options .......................................... 34
  4.3.1. Set Up the Analyses Time and Data Location ................................... 34
  4.3.2. Set Up Analysis Algorithm .................................................................. 34
  4.3.3. Set Up the Analyses Variables ......................................................... 35
  4.3.4. Set Up the Ensemble Backgrounds ..................................................... 35
  4.3.5. Set Up Localization Distances ........................................................... 36
  4.3.6. Set Up Adaptive Posterior Inflation Parameters .................................... 37
  4.3.7. Satellite Observations Related Parameters ......................................... 38
  4.3.8. Observation QC Parameters .............................................................. 38

5. Applications for Regional and Global EnKF .............................................. 39
  5.1. Running GSI Observer for Regional Applications ...................................... 42
    5.1.1. Run Script ...................................................................................... 42
    5.1.2. Run GSI Observer and Check Run Status ......................................... 44
    5.1.3. Check for Successful GSI Completion .............................................. 46
  5.2. Running EnKF for Regional Applications ............................................... 48
    5.2.1. Run Script ...................................................................................... 48
    5.2.2. Run EnKF and Check Run Status .................................................... 49
    5.2.3. Check for Successful EnKF Completion ............................................ 51
    5.2.4. Diagnose EnKF Analysis Results .................................................... 53
  5.3. Running GSI Observer for Global Applications ......................................... 54
    5.3.1. Run Script ...................................................................................... 54
    5.3.2. Run GSI Observer and Check Run Status ......................................... 57
    5.3.3. Check for Successful GSI Completion .............................................. 59
  5.4. Running EnKF for Global Applications .................................................. 60
    5.4.1. Run Script ...................................................................................... 60
    5.4.2. Run EnKF and Check Run Status .................................................... 62
    5.4.3. Check for Successful EnKF Completion ............................................ 63
    5.4.4. Diagnose EnKF Analysis Results .................................................... 65

6. EnKF Basic Concepts and Code Structure ................................................. 68
  6.1. Basic Concepts (in the Form of EnSRF) .................................................. 68
    6.1.1. Analysis Variables ........................................................................... 68
    6.1.2. Update of Analysis Variables .......................................................... 69
    6.1.3. Updates of Observation Priors ......................................................... 70
    6.1.4. Assimilation Order and Adaptive Thinning of Observations ................ 71
    6.1.5. Ensemble Spread Inflation .............................................................. 71
    6.1.6. Covariance Localization ................................................................. 72
    6.1.7. Adaptive Radiance Bias Correction with EnSRF ................................ 73
  6.2. EnSRF Code Structure and Key Functions ............................................... 73
    6.2.1. Main Code Tree .............................................................................. 73
    6.2.2. Driver of Serial Ensemble Square Root Filter .................................... 74

A. Content of Namelist ..................................................................................... 76

Bibliography ..................................................................................................... 81
Overview

1.1 EnKF History and Background

The ensemble Kalman filter (EnKF) is a Monte-Carlo algorithm for data assimilation that uses an ensemble of short-term forecasts to estimate the background-error covariance in the Kalman filter. Each ensemble member is cycled through the data assimilation system and updated by EnKF. The EnKF code was developed by the National Oceanic and Atmospheric Administration (NOAA) Earth System Research Lab (ESRL) in collaboration with the research community. It contains two separate algorithms for calculating an analysis increment, a serial Ensemble Square Root Filter (EnSRF) algorithm described by [6] and a Local Ensemble Kalman Filter (LETKF) algorithm described by [4]. The parallelization scheme used by the EnSRF algorithm is based on that used in the Data Assimilation Research Testbed (DART) toolkit developed at the National Center for Atmospheric Research (NCAR) and described by [1]. The LETKF code was contributed by Yoichiro Ota of the Japanese Meteorological Agency (JMA) while he was a visitor at the National Centers for Environmental Prediction (NCEP).

The EnKF code became an operational data assimilation system at NCEP in May 2012, providing ensemble update to the Global Forecast System (GFS) hybrid Ensemble-Variational (EnVar) data assimilation system. It can work with both global forecast models (e.g., GFS) and regional forecast models (e.g., the Hurricane Weather Research and Forecasting (WRF) (HWRF) model, the North American Mesoscale (NAM) system, the Advanced Research WRF (ARW) system).
1. Overview

1.2 EnKF Becomes Community Code

The Developmental Testbed Center (DTC), in collaboration with major development groups, began transforming the EnKF operational system into a community system in 2014, following the same protocol as the GSI community effort ([5], http://www.dtcenter.org/com-GSI/users/). Consequently, the EnKF code and its user support are managed by the DTC, along with the GSI community system.

The DTC complements the development groups in providing EnKF documentation, porting EnKF to multiple platforms, and testing EnKF in an independent and objective environment, while still maintaining functionally equivalent to operational centers. Working with code developers, the DTC is maintaining a unified community GSI/EnKF repository to facilitate community users to develop EnKF. Based on the repository, the DTC releases EnKF code annually with GSI. The first community version of the EnKF system was released on July 31, 2015. This user’s guide describes the third release of EnKF (v1.3) in November 2018. The DTC provides user support through the EnKF Helpdesk (enkf-help@ucar.edu), tutorials, and workshops. More information about the EnKF community services can be found at the DTC EnKF webpage (http://www.dtcenter.org/EnKF/users/).

1.2.1 EnKF Code Management and Review Committee

The EnKF code development and maintenance are administrated by the Data Assimilation Review Committee (DARC). DARC was originally formed as the GSI Review Committee in 2010. The committee was reformed in 2014 to include members representing the EnKF development and applications. Such a combination enhanced collaboration of development groups in both variational and ensemble data assimilation communities. Currently, DARC contains members from NCEP’s Environmental Modeling Center (EMC), the National Aeronautics and Space Administration (NASA) Goddard Global Modeling and Assimilation Office (GMAO), NOAA/ESRL, the National Center for Atmospheric Research (NCAR) Mesoscale & Microscale Meteorology Laboratory (MMM), the National Environmental Satellite, Data, and Information Service (NESDIS), the United States Air Force (USAF), the University of Maryland, Joint Center for Satellite Data Assimilation (JCSDA), and the DTC.

DARC primarily steers distributed GSI/EnKF development and community code management and support. The responsibilities of the committee are divided into two major aspects: coordination and code review. The purpose and guiding principles of the review committee are described in the Chapter I of the GSI User’s Guide.

1.2.2 Community Code Contributions

EnKF is a community data assimilation system, open to contributions from scientists and software engineers from both the operational and research communities. DARC oversees the
code transition from prospective contributors. This committee reviews proposals for code
commits to the GSI/EnKF repository and monitors that coding standards and tests are being
fulfilled. Once the committee reaches approval, the contributed code will be committed
to the GSI/EnKF code repository and available for operational implementation and public
release.

To facilitate this process, the DTC is providing code transition assistance to the general
research community. Prospective contributors should contact the DTC EnKF helpdesk
(enkf-help@ucar.edu) as early as possible. It is the contributor's responsibility to ensure
that a proposed code change is correct, meeting the EnKF coding standards, and well-
documented. The DTC will help the contributors run the regression tests and merge
the code with the top of the repository. Prospective contributors can also apply to the
DTC visitor program for their EnKF research and code transition. The visitor pro-
gram is open to applications year-round. Please check the visitor program webpage
(https://dtcenter.org/visitor-program) for the latest announcement of opportunity
and application procedures.

1.3 About This EnKF Release

This user's guide was composed for the EnKF community release version(v) 1.3, which is
compatible with the GSI community release v3.7. Please note the major focuses of the DTC
are currently on testing and evaluation of EnKF for regional numerical weather predic-
tion (NWP) applications though the instructions and cases for EnKF global applications are
available with this release.

Running this EnKF system requires running GSI as a prior for its observation operators.
Therefore, the GSI User's Guide is referred throughout this documentation. This GSI user's
guide can be obtained at the GSI user's webpage (http://www.dtcenter.org/com-GSI/
users/docs/index.php).

1.3.1 What Is New in This Release Version

The following lists some of the new functions and changes included in the v1.3 release of the
EnKF versus v1.2:

- Added the capability to perform model-space vertical localization using modulated
  ensembles
- Added option for GSI to write out diag files in NetCDF format as well as binary format
- Added option for EnKF to read NetCDF format diag files
- Added option of running GSI in the observer mode once for the ensemble mean and
  saving the observation operator jacobian in diag files. The jacobian is then used in
  EnKF for computing linearized observation operator for ensemble perturbations
- Added option for EnKF to save ensemble spread in diag files
1. Overview

• Bug fixes

1.3.2 Observations Used by This Version

EnKF is using the GSI system as the observation operator to generate observation innovations. Therefore, the observation types assimilated by EnKF are the same as GSI. This version of EnKF has been tested to work with the community GSI release v3.7. It can assimilate, but is not limited to, the following types of observations:

**Conventional observations (including satellite retrievals):**

- Radiosondes
- Pilot balloon (PIBAL) winds
- Synthetic tropical cyclone winds
- Wind profilers: USA, Jan Meteorological Agency (JMA)
- Conventional aircraft reports
- Aircraft to Satellite Data Relay (ASDAR) aircraft reports
- Meteorological Data Collection and Reporting System (MDCRS) aircraft reports
- Dropsondes
- Moderate Resolution Imaging Spectroradiometer (MODIS) IR and water vapor winds
- Geostationary Meteorological Satellite (GMS), JMA, and Meteosat cloud drift IR and visible winds
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT) and GOES water vapor cloud top winds
- GEOS hourly IR and cloud top wind
- Surface land observations
- Surface ship and buoy observation
- Special Sensor Microwave Imager (SSMI) wind speeds
- Quick Scatterometer (QuikSCAT), the Advanced Scatterometer (ASCAT) and Oceansat-2 Scatterometer (OSCAT) wind speed and direction
- RapidScat observations
- SSM/I and Tropical Rainfall Measuring Mission (TRMM) Microwave Imager (TMI) precipitation estimates
- Velocity-Azimuth Display (VAD) Next Generation Weather Radar ((NEXRAD) winds
- Global Positioning System (GPS) precipitable water estimates
- Sea surface temperature (SST)
- Doppler wind Lidar
- Aviation routine weather report (METAR) cloud coverage
- Flight level and Stepped Frequency Microwave Radiometer (SFMR) High Density Observation (HDOB) from reconnaissance aircraft
- Tall tower wind

**Satellite radiance/brightness temperature observations (instrument/satellite ID):**

- SBUV: *NOAA-17, NOAA-18, NOAA-19*
1. Overview

- High Resolution Infrared Radiation Sounder (HIRS): *Meteorological Operational-A (MetOp-A), MetOp-B, NOAA-17, NOAA-19*
- GOES imager: *GOES-11, GOES-12*
- Atmospheric IR Sounder (AIRS): *aqua*
- AMSU-B: *MetOp-B, NOAA-17*
- SSMI: *DMSP F14, F15, F19*
- SSMI/S: *DMSP F16*
- Advanced Microwave Scanning Radiometer for Earth Observing System (AMSR-E): *aqua*
- GOES Sounder (SNDR): *GOES-11, GOES-12, GOES-13*
- Infrared Atmospheric Sounding Interferometer (IASI): *MetOp-A, MetOp-B*
- Global Ozone Monitoring Experiment (GOME): *MetOp-A, MetOp-B*
- Ozone Monitoring Instrument (OMI): *aura*
- Spinning Enhanced Visible and Infrared Imager (SEVIRI): *Meteosat-8, Meteosat-9, Meteosat-10*
- Advanced Technology Microwave Sounder (ATMS): *Suomi NPP*
- Cross-track Infrared Sounder (CrIS): *Suomi NPP*
- GCOM-W1 AMSR2
- GPM GMI
- Megha-Tropiques SAPHIR
- Himawari AHI

Others:

- GPS Radio occultation (RO) refractivity and bending angle profiles
- Solar Backscatter Ultraviolet (SBUV) ozone profiles, Microwave Limb Sounder (MLS) (including NRT) ozone, and Ozone Monitoring Instrument (OMI) total ozone
- Doppler radar radial velocities
- Radar reflectivity Mosaic
- Tail Doppler Radar (TDR) radial velocity and super-observation
- Tropical Cyclone Vitals Database (TCVital)
- Particulate matter (PM) of 10-um diameter, 2.5-um diameter or less
- MODIS AOD (when using GSI-chem package)

Please note some of these above mentioned data are not yet fully tested and/or implemented for operations. Therefore, the current GSI code might not have the optimal setup for these data.
Software Installation

2.1 Introduction

The DTC community EnKF is a community distribution of NOAA's operational ensemble Kalman filter. The community EnKF expands the portability of the operational code by adding a flexible build system and providing example run scripts that allow the system to run on many common platforms. The current version of the community EnKF builds and runs on most standard Linux platforms using the Intel, PGI, and GNU compilers.

This chapter describes how to build and install the DTC community EnKF on your local Linux computing resources. By default EnKF is built as part of the GSI build. The only difference between the steps necessary to build GSI and EnKF are that for EnKF one of three IO configurations must be chosen prior to enacting the build.

The EnKF build process consists of the three steps:

- Obtaining the source code for EnKF/GSI.
- Setting the appropriate environment variables for the build.
- Configuring and compiling the source code using CMake.

Section 2.2 describes how to obtain the source code from the DTC website. Section 2.2.1 describes the three different IO options that can be selected prior to building the code. Section 2.3 presents an outline of the build process. Section 2.3.1 illustrates an Intel build in detail, and discusses optional arguments. Section 2.3.2 describes all of the current optional settings for the build. Sections 2.4 and 2.5 cover the system requirements (tools, libraries, and environment variable settings) and currently supported platforms in detail. Section 2.6 discusses what to do if you have problems with the build and where to get help.
2. Software Installation

2.2 Obtaining the Source Code

The community EnKF code and the GSI code are released as a combined source code package. The current EnKF release is v1.3 and is paired with the community GSI release version 3.7. The community EnKF release is available from the DTC community EnKF users website;

http://www.dtccenter.org/EnKF/users/index.php

The community GSI/EnKF release includes the source code for both the EnKF v1.3 and the GSI v3.7 models, as well as an integrated build system, utilities, and documentation necessary to build and run the EnKF.

To download the source code from either the GSI or the EnKF website, select the Download tab along with the GSI/EnKF System sub-tab on the vertical menu located on the left side of the main page. New users must first register before downloading the source code. Returning users only need to enter their registration email address to log in. After accessing the download page, select the link to the comGSIv3.7_EnKFv1.3 tarball. Please only use the source code provided with the comGSIv3.7_EnKFv1.3 tarball. Do not mix and match this tarball with other versions of the community GSI code or supplemental libraries, as this will lead to unpredictable results.

The community EnKF version 1.3 comes in a tar file named comGSIv3.7_EnKFv1.3.tar. The tar file may be unpacked by using the standard UNIX commands:

```
    gunzip comGSIv3.7_EnKFv1.3.tar.gz
    tar -xvf comGSIv3.7-EnKFv1.3.tar
```

This creates the top level GSI directory comGSIv3.7_EnKFv1.3/. After downloading the source code, and prior to building, the user should check the known issues link on the download page of DTC website to determine if any bug fixes or platform specific customizations are needed.

Specifics on the source code directory structure can be found in the GSI User's Guide section 2.3, at the DTC community GSI users website:

http://www.dtccenter.org/com-GSI/users/index.php

2.2.1 Versions of EnKF

The EnKF code has three build time configurations for the I/O; WRF regional, GFS global, and NMMB. The EnKF analysis is identical in each case, only the capability to digest model input differs. The regional version can only digest WRF formatted I/O files. The global
version can only digest spectral input from the NCEP global model, and lastly, the NMMB version can only digest NMMB files.

The default build constructs the regional configuration of EnKF. To get global EnKF, it is necessary to add optional build arguments to the cmake command.

- `-DENKF_MODE=GFS` to get the GFS version EnKF

Similar arguments control the enacting the NMMB bindings. This will be explained in detail in section 2.3.1 and 2.3.2.

### 2.3 Compiling EnKF

A new unified build system based on CMake has been added to the EnKF/GSI code. While the new build system is a significant departure from the previous DTC build system, CMake is a very powerful cross-platform open-source build system that has many capabilities to make building easier.

The CMake build infrastructure consists of a top level directory with the name `cmake/` and configuration files in each directory named (`CMakeLists.txt`). This replaces the old arch/ directory. CMake relies on a two step command line process, similar to the old UNIX "configure" and "compile." Command line arguments are used to specify paths and compilers. By default CMake is configured to build the source code "out-of-place," meaning that it does not populate the source code directory with the build. The location for the build must be specified by the user.

Once a compiler has been chosen, CMake generates local makefiles by invoking the cmake command with the proper arguments. One of those arguments selects that a local build of the NCEP libraries needed by EnKF/GSI will be conducted prior to the source code being built. The final step is to invoke a parallel build (make) of the code.

A significant advantage with using CMake to build the code, is that CMake automatically generates code dependencies each time a build is invoked, allowing the use of a parallel make as the final step of the build. This greatly reduces the time it take to complete the build. Because of the parallel compiling, the time to complete the CMake build is typically a quarter of the time needed for the serial DTC "configure" and "compile" to complete.

Once the build is complete, the EnKF executable is placed in the bin directory of the build directory. Note that the name of the executables and their location differs from the traditional DTC build.

Summary of CMake build steps:

1. Set up the build environment
   - Select a specific compiler by loading modules or setting paths as needed
   - Set the environment path for the NetCDF library (The HDF5 library is also needed if your NetCDF version is 4+).
2. Software Installation

- Set the LAPACK/MKL mathematics libraries as needed.
- Create a directory for your build and change into it.
- Run the cmake command with the appropriate arguments.
- Run make
- The executables are located in the build directory within the bin directory.

2.3.1 Example build using the Intel Fortran Compiler

This section details how to build EnKF on the NCAR supercomputer Cheyenne, using the Intel V18.0.1 fortran compiler, the 2018 MKL libraries, NetCDF v4.6.1, and the 2018 Intel MPI libraries.

- Set up your build environment

  module purge
  module load cmake/3.9.1
  module load intel/18.0.1 ncarenv/1.2 ncarcompilers/0.4.1 mkl/2018.0.1
  module load netcdf/4.6.1
  module load impi/2018.1.163

- Next, create a build directory and move into it.

  mkdir ./build
  cd build

- From within the build directory, invoke the cmake command to build the global GFS version of EnKF.

  cmake -DENKF_MODE=GFS path_to_comGSI_directory

  The makefiles customized to your platform and the environment, are now created.

- Run make (parallel compile) on eight cores

  make -j8

  Because CMake has already figured out the dependencies, the make can run in parallel.

Let's look at relevant arguments for cmake when building EnKF. The first is `DBUILD_CORELIBS=ON`. This argument directs CMake to look in the libsrc/ directory to build the NCEP libraries needed for EnKF and GSI.

The argument `CMAKE_C_COMPILER` specifies the C compiler to use with the build. For the Intel compiler, use `icc`. The argument `CMAKE_CXX_COMPILER` specifies the C++ compiler to use with the build. For the Intel compiler, use `icpc`. The argument `CMAKE_Fortran_COMPILER` specifies the Fortran compiler to use with the build. Because this is an parallel MPI code, on Cheyenne use `mpif90` instead of `ifort`. Typically, the only time one would explicitly specify the compilers on the command line is when you wish to override a default choice set in your dot files or if you were to create a batch script to conduct the build.

To run the cmake command with a compiler other than Intel only requires replacing the values for `CMAKE_C_COMPILER`, `CMAKE_CXX_COMPILER`, and `CMAKE_Fortran_COMPILER` that correspond to the new compiler. Table 2.1 lists these values.
2. Software Installation

Table 2.1: The compiler specific values for the cmake compiler argument

<table>
<thead>
<tr>
<th>Compiler</th>
<th>C</th>
<th>C++</th>
<th>FC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel</td>
<td>icc</td>
<td>icpc</td>
<td>mpif90</td>
</tr>
<tr>
<td>PGI</td>
<td>pgcc</td>
<td>pgcc++</td>
<td>mpif90</td>
</tr>
<tr>
<td>GNU</td>
<td>gcc</td>
<td>g++</td>
<td>mpif90</td>
</tr>
</tbody>
</table>

2.3.2 Default Options

The CMake build has a number of options to when building. These are enacted by adding the option with the prefix -D to the command line call to cmake. For instance, in the example above, the BUILD_CORELIBS option is enacted by adding -DBUILD_CORELIBS=ON to the cmake call. The currently available command line build options are:

- **BUILD_ENKF**: builds the enkf executable (default is ON)
- **BUILD_CORELIBS**: Attempts to build the core libraries from source
- **USE_WRF**: Builds GSI with WRF dependencies (default is ON)
- **BUILD_GFS**: Builds ENKF using GFS (default is ON)
- **BUILD_WRF**: Builds ENKF using WRF (default is OFF)
- **BUILD_NMMB**: Builds ENKF using NMMB (default is OFF)

Note that by default EnKF built is global with GFS. See the readme file `README.cmake` in the top directory for a current list of options; or look in the CMakeLists.txt in the top directory and search for options.

2.4 System Requirements and Compiler Specific Notes

The EnKF source code is written in FORTRAN, FORTRAN 90, and C. In addition, the parallel executables require some flavor of MPI and OpenMP for the distributed memory parallelism. The I/O relies on the NetCDF I/O libraries. And CMake is needed to build the code.

The basic requirements for building and running the GSI system are the following:

- CMake V2.8+
- A FORTRAN compiler that supports the 2003 or newer standard
- C compiler
- MPI v1.2+
- OpenMP
- NetCDF V4.2+ and HDF5
- LAPACK and BLAS mathematics libraries, or equivalent

In addition, GSI and EnKF require linear algebra libraries such as LAPACK and BLAS. The Intel compiler usually comes with a vendor provided mathematics library known as the *Mathematics Kernel Libraries* or MKL for short. While most installations of the Intel compiler
come with the MKL libraries installed not all do. In addition, if it is not installed correctly, the ifort compiler does not automatically load the library. It is therefore necessary to set the LAPACK_PATH variable to the location of the MKL libraries when using the Intel compiler. You may need to ask your system administrator for the correct path to these libraries.

IBM systems typically come installed with the LAPACK equivalent ESSL library that links automatically. Likewise, the PGI compiler often comes with a vendor provided version of LAPACK that links automatically with the compiler. The Gnu compiler is the only one that does not come with any installed version of the LAPACK and BLAS libraries.

Because all but the last of these tools and libraries are typically the purview of system administrators to install and maintain, they are lumped together here as part of the basic system requirements.

2.5 Compilers Tested for Release

Version 1.3 of the DTC community EnKF system has been extensively tested with a variety of compilers and versions of those compilers, on standard Linux platforms. To summarize, Intel compiler versions 16-18 should build and run without issue. Early version 15 has some known issues, but 15.1, 15.3, and 15.6 should work. For the Portland Group compiler, versions 16-18 should work without issue, and Gnu version 6.3, 7.1, and 7.3 are known to work. Version 18 of Gnu has issues and is not currently supported. See table ?? for the complete list compilers and versions tested. See release notes (https://dtcenter.org/com-GSI/users/support/release_notes/index_GSIv3.7.php) for the complete list compilers and versions tested.

Unforeseen build issues may occur when using older compiler and library versions. As always, the best results will be achieved by using the most recent compiler versions.

2.6 Getting Help and Reporting Problems

Should the user experience any difficulty building EnKF on their system, please first confirm that GSI has successfully built. Feel free to contact the community EnKF support at

enkf-help@ucar.edu

for assistance.

At a minimum, when reporting issues building the code, please include a copy of the EnKF build log.
Running EnKF

3.1 Input Data Required to Run EnKF

This chapter discusses the process of running EnKF cases. It includes:

1. Discussions of the input data required to run EnKF.
2. A detailed explanation of how to run both a regional and global EnKF with the released run scripts.
3. Introduction to the files in a successful regional and global EnKF run directory

3.1.1 Input Data Required to Run EnKF

In most cases, three types of input data, including ensemble mean and members, observations, and fixed files, must be available before running EnKF.

1. Ensemble mean and members
The ensemble members and ensemble mean of certain regional and global ensemble forecast systems are used as the background for the EnKF analysis. When this EnKF system is used for regional analysis with WRF ensembles, the ensemble members and ensemble mean follow the naming convention:

firstguess.mem001
firstguess.mem002
... ...
3. Running EnKF

Please note that the number of allocated computer cores to run EnKF must be larger than the ensemble size. The ensemble members can be generated using various methods, such as:

- Using global/regional ensemble forecasts.
- Ensemble forecasts generated using multi-physics, multi-models, or adding random perturbations drawn from climatology.
- In cycling assimilation, using ensemble forecasts initialized from previous ensemble analyses generated by EnKF.

This version EnKF can use any of the following ensemble files as the background:

- ARW NetCDF forecast
- NMM or NMMB forecast (not tested and supported by DTC)
- GFS forecast

2. Prepare observation ensemble priors (observation innovation)

In addition to the ensemble backgrounds on model grids, the ensemble priors of all observations (observation innovation for all ensemble members) are also needed to run EnKF. The observation ensemble priors are generated by running GSI observation forward operators with the ensemble members as backgrounds (without doing actual GSI analyses). In this release, the GSI v3.7 run script includes options for generating the observation ensemble priors (details in section 3.2). The observation ensemble priors files should follow the following naming conventions:

- For conventional observations:
  
  diag_conv_ges.mem001
diag_conv_ges.mem002
  ...
  diag_conv_ges.ensmean

- For radiance observations:

  diag_instrument_Satellite.mem001 (e.g. diag_hirs4_n19.mem001)
diag_instrument_Satellite.mem002 (e.g. diag_hirs4_n19.mem002)
  ...
  diag_instrument_Satellite.ensmean (e.g. diag_hirs4_n19.ensmean)

These diag files contain a lot of information about each observation. For more details on the content of diag files, please refer to the GSI User’s Guide Appendix A.2. The preparation of observations for EnKF assimilation is done within GSI, including quality control of observations, selection of observation types for assimilation, and observation error tuning. In the default namelist situation, NO additional online quality control of observations is performed in the EnKF analysis step (although there is an option to do the similar quality control of observations as the GSI variational scheme.)

3. Fixed files

EnKF uses the the same fixed files as GSI to setup the analysis configurations. Detailed
3. Running EnKF

Explaination of all fixed files provided in the community GSI system is in the GSI user's Guide, Chapter 3. The following is a list of fixed files needed for EnKF analyses:

- for observation control:
  
  - convinfo - conventional data (prepufr) info file
  - ozinfo - ozone retrieval info file
  - satinfo - satellite channel info file

- when satellite radiance data are assimilated, the following files are needed to do the adaptive radiance bias correction:
  
  - satbias_in - satellite bias correction coefficient file
  - satbias_pc - satellite bias correction coefficient file for passive channels

Note that this version EnKF uses adaptive bias correction. The bias correction coefficients are in a single file that combined satellite angle dependent and mass bias correction coefficients. See GSI User's Guide for more detail. When the namelist parameter `readin_localization` is set to true, file `hybens_locinfo` is needed, in which customized localization values varying by model level are contained.

3.2 EnKF and GSI Observer Run Scripts

In this release version, four sample run scripts for EnKF applications are under directory `comGSIv3.7_EnKFv1.3/ush`:

- `ush/comgsi_run_regional.ksh` for running regional GSI to generate the observation ensemble priors. Referred to as the GSI observer run scripts.
- `ush/comgsi_run_global.ksh` is GSI observer run script for global applications (to generate the observation ensemble priors).
- `ush/comenkf_run_regional.ksh` for running regional EnKF
- `ush/comenkf_run_global.ksh` for running global EnKF

These run scripts are introduced in detail in the following sections. Also provided are three scripts for generating the GSI and EnKF namelist:

- `ush/comgsi_namelist.sh` generates GSI namelist on the fly (called by GSI observer run scripts).
- `ush/comgsi_namelist_gfs.sh` generates GSI namelist on the fly (called by GSI observer run scripts) for global GSI applications.
- `ush/comenkf_namelist.sh` generates EnKF namelist on the fly (called by EnKF run script) for regional EnKF applications.

3.2.1 General Introduction to the Run Scripts

These run scripts provide the run time environment necessary for running the GSI and EnKF executables. They all have similar steps, as follows:
3. Running EnKF

1. Request computer resources to run GSI/EnKF.
2. Set environmental variables for the machine architecture.
3. Set experimental variables (such as experiment name, analysis time, background, and observation).
4. Check the definitions of required variables. Generate a run directory for GSI/EnKF
5. Copy the GSI/EnKF executable to the run directory.
6. Copy/Link the background file/ensemble files to the run directory.
7. Link observations to the run directory.
8. Link fixed files (statistic, control, and coefficient files) to the run directory.
9. Generate namelist for GSI/EnKF.
10. Run the GSI/EnKF executable.

In the GSI User’s Guide, three sections explain the first three steps in detail:

- Section 3.2.2.1: Setting up the machine environment (step 1)
- Section 3.2.2.2: Setting up the running environment (step 2)
- Section 3.2.2.3: Setting up an analysis case (step 3)

For this documentation, the first 2 steps will be skipped and the 3rd step in the GSI observer and EnKF run scripts will be discussed. The community GSI analysis run script and the GSI observer run script are the same, with the GSI observer capability controlled by flags that turn off the minimization, select appropriate namelist options and enable looping through all the ensemble members to generate the ensemble observation priors for each member, including the ensemble mean.

3.2.2 GSI Observer Run Scripts

Setting up a case

This section discusses variables specific to the user’s case, such as analysis time, working directory, background and observation files, location of fixed files and CRTM coefficients, and the GSI executable file. The script looks like:

```
# case setup (users should change this part)
# ANAL_TIME= analysis time (YYYYMMDDHH)
# WORK_ROOT= working directory, where GSI runs
# PREPBURF = path of PreBUFR conventional obs
# BK_FILE = path and name of background file
# OBS_ROOT = path of observations files
# FIX_ROOT = path of fix files
# GSI_EXE = path and name of the gsi executable
# ENS_ROOT = path where ensemble background files exist

ANAL_TIME=2017051318
JOB_DIR=the_job_directory
RUN_NAME=a_descriptive_run_name_such_as_case05_3denvar_etc
OBS_ROOT=the_directory_where_observation_files_are_located
BK_ROOT=the_directory_where_background_files_are_located
```
### 3. Running EnKF

```bash
GSI_ROOT=the_comgsi_main_directory where src/ ush/ fix/ etc are located
CRTM_ROOT=theCRTM_directory
ENS_ROOT=the_directory where ensemble backgrounds are located

#ENS_ROOT is not required if not running hybrid EnVar

HH='echo $ANAL_TIME | cut -c9-10'
GSI_EXE=${JOB_DIR}/gsi.x #assume you have a copy of gsi.x here
WORK_ROOT=${JOB_DIR}/${RUN_NAME}
FIX_ROOT=${GSI_ROOT}/fix
GSI_NAMELIST=${GSI_ROOT}/ush/comgsi_namelist.sh
PREPBURF=$(OBS_ROOT)/nam.t$HHz.prepbufr.tm00
BK_FILE=${BK_ROOT}/wrfinput_d01.$ANAL_TIME

The options ANAL_TIME, WORK_ROOT, PREPBURF, BK_FILE, OBS_ROOT, FIX_ROOT, GSI_EXE are all the same settings as the GSI analysis configuration. Two options: BK_ROOT, GSI_ROOT, are the root directories for ensemble members and the GSI system. These exist to make links to the background and GSI system easy and shorter. The new option: GSI_NAMELIST is needed because the namelist section was taken out of the run scripts in this release as a separate file to improve the structure and readability of the run scripts. Users can find the namelist files for both GSI and EnKF in the same directory as the run scripts. Please note the option BK_FILE is pointing to the ensemble mean. The next part of this block has additional options to specify other important aspects of the GSI observer.
```

```bash
#------------------------------------------------
# bk_core= which WRF core is used as background (NMM or ARW or NMMB)
# bkcv_option= which background error covariance and parameter will be used
# (GLOBAL or NAM)
# if_clean = clean : delete temperal files in working directory (default)
# no : leave running directory as is (this is for debug only)
# if_observer = Yes : only used as observation operator for enkf
# if_hybrid = Yes : Run GSI as 3D/4D EnVar
# if_4DEnVar = Yes : Run GSI as 4D EnVar
# if_nemsio = Yes : The GFS background files are in NEMSIO format
# if_oneob = Yes : Do single observation test
if_hybrid=No # Yes, or, No -- case sensitive !
if_4DEnVar=No # Yes, or, No -- case sensitive !
if_nemsio=No # Yes, or, No -- case sensitive !
if_oneob=No # Yes, or, No -- case sensitive !

bk_core=ARW
bkcv_option=NAM
if_clean=clean

... # The following two only apply when if_observer = Yes, i.e. run observation operator for EnKF
no_member number of ensemble members
BK_FILE_mem path and base for ensemble members

The options bk_core, bkcv_option, and, if_clean are the same as in the GSI analysis run scripts. The new option if_observer indicates if this GSI run is for the generation of the observation ensemble priors or for a regular GSI run. The new option no_member specifies the number of ensemble members that are need to be calculated for the observation ensemble priors. This should also be the ensemble number in the EnKF analysis. Option BK_FILE_mem is the path and the name of the ensembles without the ensemble member ID appended. The scripts will add the ensemble member ID as a three digital number, such as 000, 001, ....
Loop through ensemble members

As mentioned previously, the GSI ensemble observer run scripts are the same as the GSI analysis run scripts released with the community GSI. Since the observer only generates diag files, which includes useful information on the observation innovation, the GSI outer loop number for the observer should be set to 0 to skip all minimization iterations.

The contents of the run scripts can be divided into two parts, those before the following comments and those after:

```
# start to calculate diag files for each member
```

Before this comment, the scripts have the same functionality as when running a GSI analysis, except that the background options in the scripts for the observer functionality are set for the ensemble mean. Additionally, the namelist is built with the two following options set, which skips the minimization and saves all observation processing from the ensemble mean:

```bash
if [ ${if_observer} = Yes ] ; then
  nummiter=0
  if_read_obs_save='.true.'
  if_read_obs_skip='.false.'
else
  Please refer to the GSI user's guide for detailed explanation of the remainder of this portion of the run scripts. The second portion of the script loops through each member to calculate the observation ensemble priors based on the GSI run environments setup by the first portion.

Listed below is an annotated version of the 2nd part of the GSI observer run script with explanations on each function block.

```bash
if [ ${if_observer} = Yes ] ; then

  # This 2nd part of the script only runs if option if observer is set to "Yes". The diag files from the ensemble mean need to be saved first with the following commands:

  string=ges
  for type in $listall; do
    count=0
    if [[ -f diag_${type}_${string}.${ANAL_TIME} ]]; then
      mv diag_${type}_${string}.${ANAL_TIME} diag_${type}_${string}.ensmean
    fi
  done
  mv wrf_inout wrf_inout_ensmean

  The following section builds the namelist for ensemble members. Please note two options need to be set different between mean and members:
```
3. Running EnKF

# Build the GSI namelist on-the-fly for each member
nummiter=0
if_read_obs_save='.false.'
if_read_obs_skip='.true.'
. $GSI_NAMELIST

The option if_read_obs_save and if_read_obs_skip switch from "True" and "False", respectively for the mean to "False" and "True", respectively for the ensemble members.

This saves all observation processing information (including bias correction, thinning, etc) from the ensemble mean and save the same information for the ensemble members to keep observations constant.

The script loops through each ensemble member (from member 001 to no_member) to create the diag files for each member:

# Loop through each member
loop="01"
ensmem=1
while [[ $ensmem -le $no_member ]];do
  rm pe0*
  print "$ensmem is $ensmem"
  ensmemid='printf %3.3i $ensmem'

  After a member is processed, the script removes the old ensemble member and links to the new member before rerunning the calculation:

# get new background for each member
if [[ -f wrf_inout ]]; then
  rm wrf_inout
fi
BK_FILE=${BK_FILE_mem}${ensmemid}
echo $BK_FILE
ln -s $BK_FILE wrf_inout

Run the GSI observer for this member:

# run GSI
echo ' Run GSI with' $(bk_core) ' for member ', $(ensmemid)
case $ARCH in
   IBM_LSF')
   $(RUN_COMMAND) ./gsi.exe < gsiparm.anl > stdout_mem$(ensmemid) 2>&1 ;;
   *) $(RUN_COMMAND) ./gsi.exe > stdout_mem$(ensmemid) 2>&1 ;;
esac
# run time error check and save run time file status
error=$?
if [ $error -ne 0 ]; then
  echo "ERROR: $GSI crashed for member $(ensmemid) Exit status=$error"
  exit $error
fi
ls -l * > list_run_directory_mem$(ensmemid)
3. Running EnKF

Generate diag files for this member:

```bash
# generate diag files
for type in $listall; do
count='ls pe*${type}_${loop}* | wc -l'
if [[ $count -gt 0 ]]; then
cat pe*${type}_${loop}* > diag_${type}_${string}.mem${ensmemid}
done
```

Since all members are using the same run directory, the run status of each member is overwritten by the following member. The stdout file and the file list in run directory are preserved with the ensemble member ID for debug.

### 3.2.3 Sample Regional EnKF Run Scripts

As described in section 3.2.1, the regional EnKF run scripts have been designed to have a similar structure to the GSI analysis and observer run scripts. Again, please refer to the GSI User’s Guide section 3.2.2.1 and 3.2.2.2 for the first two steps.

The 3rd step is to setup the variables specific to the user’s case, such as analysis time, working directory, background and observation files, location of fixed files and CRTM coefficients, and the EnKF executable. Most of the options in this portion are the same as the 3rd step in the GSI observer run scripts, which is discussed in section 3.2.2 of this User’s Guide. Users should setup most of the variables in this portion based on the options in the GSI observer run scripts as they are the variables to setup the same things for the GSI and EnKF. The following is a sample script with explanations:

```
# case set up (users should change this part)
# ANAL_TIME= analysis time (YYYYMMDDHH)
# WORK_ROOT= working directory, where GSI runs
# PREPBURF = path of PreBUFR conventional obs
# OBS_ROOT = path of observations files
# FIX_ROOT = path of fix files
# ENKF_EXE = path and name of the EnKF executable
ANAL_TIME=2014021300 #used by comenkf_namelist.sh
JOB_DIR=the_job_directory
RUN_NAME=a_descriptive_run_name_such_as_case05_3denvar_etc
OBS_ROOT=the_directory_where_observation_files_are_located
BK_ROOT=the_directory_where_background_files_are_located
GSI_ROOT=the_comgsi_main_directory_where_src_ush_fix_etc_are_located
CRTM_ROOT=the_CRTM_directory
diag_ROOT=the_observer_directory_where_diag_files_exist
ENKF_EXE=${JOB_DIR}/enkf_wrf.x
WORK_ROOT=${GSI_ROOT}/fix
ENKF_NAMELIST=${GSI_ROOT}/ush/comenkf_namelist.sh
```
3. Running EnKF

Options $ANAL\_TIME$, $BK\_ROOT$, $BK\_FILE$, $GSI\_ROOT$, $FIX\_ROOT$, $CRTM\_ROOT$ have the same meanings as the ensemble GSI observer run scripts and should be set to the same values as the ensemble GSI observer run scripts. The option $WORK\_ROOT$ is the working directory which should have enough space to hold the ensemble members and EnKF analysis results. The option $diag\_ROOT$ is pointing to the run directory of the GSI observer, where the diag files are generated as data input for the EnKF. The option $ENKF\_EXE$ points to the EnKF executable, which is under the GSI source code directory in this release. The option $ENKF\_NAMELIST$ is the path and the EnKF namelist file, which sits outside of the run script as a separate file like the GSI namelist. Users can find the namelist files for both GSI and the EnKF in the same directory as the run scripts.

The next part of this block includes several additional options that specify several aspects of the ensemble members.

```
# ensemble parameters
#
 NMEM\_ENKF=20
 BK\_FILE\_mem=${BK\_ROOT}/wrfarw
 NLONS=129
 NLATS=70
 NLEVS=50
 IF\_ARW=.true.
 IF\_NMM=.false.
 list="conv"
 # list="conv amsua\_n18 mhs\_n18 hirs\_n19"
```

Options $NMEM\_ENKF$, $BK\_FILE\_mem$ are also in the GSI observer run script and should be set to the same values as the GSI observer run script. Options $NLONS$, $NLATS$, and $NLEVS$ specify 3 dimensions (XYZ) of the ensemble grid. Options $IF\_ARW$ and $IF\_NMM$ indicates which background, ARW NetCDF ensemble or the NMM NetCDF ensemble, is used in this EnKF run. Option $list$ is a list of the observation types that the EnKF will use in the analysis. This list should be based on the diag files generated by the ensemble GSI observer.

At this point, users should be able to run the EnKF for simple cases without changing the rest of the script. However, some advanced users may need to change some of the following blocks for special applications.

```
###########################################
# Users should NOT change script after this point
###########################################
```

The next block sets the run command to run EnKF on multiple platforms. The $ARCH$ is set at the beginning of the script.

```
case $ARCH in
 'IBM\_LSF')
   ###### IBM LSF (Load Sharing Facility)
   RUN\_COMMAND="mpirun\_lsf " ;;

 'LINUX')
   if [ $GSIPROC = 1 ]; then
      ###### Linux workstation - single processor
      RUN\_COMMAND=""
   else
      ###### Linux workstation - mpi run
```

20
3. Running EnKF

```bash
RUN_COMMAND="mpirun -np $(GSIPROC) -machinefile ~/mach "
fi ;;

'LINUX_LSF'
    ####### LINUX LSF (Load Sharing Facility)
    RUN_COMMAND="mpirun.lsf " ;;

'LINUX_PBS'
    ####### Linux cluster PBS (Portable Batch System)
    # RUN_COMMAND="mpirun -np $(GSIPROC) " ;;
    RUN_COMMAND="mpiexec_mpt -n $(GSIPROC) " ;;

'DARWIN_PGI'
    ####### Mac - mpi run
    if [ $GSIPROC = 1 ]; then
        ####### Mac workstation - single processor
        RUN_COMMAND=""
    else
        ####### Mac workstation - mpi run
        RUN_COMMAND="mpirun -np $(GSIPROC) -machinefile ~/mach "
    fi ;;
    *
    print "error: $ARCH is not a supported platform configuration."
    exit 1 ;;
esac
```

The following block sets up fixed files and some analysis-time related values:

```bash
# Given the analysis date, compute the date from which the
# first guess comes. Extract cycle and set prefix and suffix
# for guess and observation data files
# gdate='$ndate -06 $adate'
gdate=$ANAL_TIME
YYYYMMDD='echo $adate | cut -c1-8'
HH='echo $adate | cut -c9-10'

# Fixed files
# CONVINFO=${FIX_ROOT}/global_convinfo.txt
# SATINFO=${FIX_ROOT}/global_satinfo.txt
# SCANINFO=${FIX_ROOT}/global_scaninfo.txt
# OZINFO=${FIX_ROOT}/global_ozinfo.txt
CONVINFO=${diag_ROOT}/convinfo
SATINFO=${diag_ROOT}/satinfo
SCANINFO=${diag_ROOT}/scaninfo
OZINFO=${diag_ROOT}/ozinfo
# LOCINFO=${FIX_ROOT}/global_hybens_locinfo.l64.txt

The next block creates a working directory (workdir) in which EnKF will run. The directory
should have enough disk space to hold all the files needed for this run. This directory is
cleaned before each run, therefore, save all the files needed from the previous run before
rerunning EnKF.

```bash
# Set up workdir
rm -rf $WORK_ROOT
mkdir -p $WORK_ROOT
cd $WORK_ROOT
```

After creating a working directory, copy or link the EnKF executable, ensembles, diag files
(observations), bias correction coefficients, and fixed files into the working directory.

```bash
cp $ENKF_EXE ./enkf.x
cp $CONVINFO ./convinfo
```
3. Running EnKF

```bash
cp $SATINFO ./satinfo
cp $SCANINFO ./scaninfo
cp $OZINFO ./ozinfo
# cp $LOCINFO ./hybens_locinfo
cp $diag_ROOT/satbias_in ./satbias_in
cp $diag_ROOT/satbias_pc ./satbias_pc

# get mean
ln -s $(BK_FILE_mem).ensmean ./firstguess.ensmean
for type in $list; do
    ln -s $diag_ROOT/diag_${type}_ges.ensmean .
done

# get each member
imem=1
while [[ $imem -le $NMEM_ENKF ]]; do
    member="mem""printf %03i $imem"
    ln -s $(BK_FILE_mem).$member ./firstguess.$member
    for type in $list; do
        ln -s $diag_ROOT/diag_${type}_ges.$member .
done
    (( imem = $imem + 1 ))
done
```

The following script is used to generate the EnKF namelist called enkf.nml in the working directory. Some namelist variables are explained in detail in Section 4.3. Appendix A gives a full list of namelist options.

```bash
# Build the GSI namelist on-the-fly
. $ENKF_NAMELIST
cat << EOF > enkf.nml
$enkf_namelist
EOF
Copy the ensemble background files to the working directory and rename them as "analysis.${member}". The EnKF will update those files as the analysis results.

```bash
# make analysis files
cp firstguess.ensmean analysis.ensmean
# get each member
imem=1
while [[ $imem -le $NMEM_ENKF ]]; do
    member="mem""printf %03i $imem"
    cp firstguess.$member analysis.$member
    (( imem = $imem + 1 ))
done
```

The following block runs EnKF and checks if the EnKF has successfully completed.

```bash
# run EnKF
#%%$RUN_COMMAND% ./enkf.x < enkf.nml > stdout 2>&1
```

```bash
# run time error check
error=$?
```
3. Running EnKF

```bash
if [ \$\{error\} -ne 0 ]; then
    echo "ERROR: \$\{ENKF_EXE\} crashed Exit status=\$\{error\}"
    exit \$\{error\}
fi
```

If this point is reached, the EnKF successfully finishes and exits with 0:

```bash
exit
```

3.3 Understanding Resulting Files in GSI Observer and EnKF Run Directory

To check if the GSI observer and EnKF runs have been successfully finished, it is important to understand the meaning of each file in the run directory.

3.3.1 The GSI Observer Run Directory

After customizing the GSI observer run script to your personal environment, it may be submitted to the batch system just as any other job. Following a successful run, the majority of the files in the GSI observer run directory will be the same as those in a successful GSI analysis run directory. The difference for the observer run is that the GSI observer generates more diag and stdout files related to each ensemble member. Below is an example of the files generated in the run directory from a GSI observer run:

```bash
amsuabufr diag_amsua_n18_ges_mem010 fit_t1.2014021300 mhsbufr
amsubbufr diag_amsua_n18_ges_mem011 fit_v1.2014021300 obs_input.0001
anavinfo diag_amsua_n18_ges_mem012 fort.201 obs_input.0002
berror_stats diag_amsua_n18_ges_mem013 fort.202 obs_input.0003
convinfo diag_amsua_n18_ges_mem014 fort.203 obs_input.0004
diag_amsua_metop-a_ges.ensmean diag_amsua_n18_ges_mem015 fort.204 obs_input.0005
diag_amsua_metop-a_ges_mem001 diag_amsua_n18_ges_mem016 fort.205 obs_input.0006
diag_amsua_metop-a_ges_mem002 diag_amsua_n18_ges_mem017 fort.206 obs_input.0007
diag_amsua_metop-a_ges_mem003 diag_amsua_n18_ges_mem018 fort.207 obs_input.0008
diag_amsua_metop-a_ges_mem004 diag_amsua_n18_ges_mem019 fort.208 obs_input.0009
diag_amsua_metop-a_ges_mem005 diag_amsua_n18_ges_mem020 fort.209 obs_input.0010
diag_amsua_metop-a_ges_mem006 diag_conv_ges.ensmean fort.210 obs_input.0011
diag_amsua_metop-a_ges_mem007 diag_conv_ges_mem001 fort.211 obs_input.0012
diag_amsua_metop-a_ges_mem008 diag_conv_ges_mem002 fort.212 obs_input.0013
diag_amsua_metop-a_ges_mem009 diag_conv_ges_mem003 fort.213 obs_input.0014
diag_amsua_metop-a_ges_mem010 diag_conv_ges_mem004 fort.214 obs_input.0015
diag_amsua_metop-a_ges_mem011 diag_conv_ges_mem005 fort.215 obs_input.0016
diag_amsua_metop-a_ges_mem012 diag_conv_ges_mem006 fort.216 obs_input.0017
diag_amsua_metop-a_ges_mem013 diag_conv_ges_mem007 fort.217 obs_input.0018
diag_amsua_metop-a_ges_mem014 diag_conv_ges_mem008 fort.218 obs_input.0019
diag_amsua_metop-a_ges_mem015 diag_conv_ges_mem009 fort.219 obs_input.0020
diag_amsua_metop-a_ges_mem016 diag_conv_ges_mem010 fort.220 obs_input.0021
diag_amsua_metop-a_ges_mem017 diag_conv_ges_mem011 fort.221 obs_input.0022
diag_amsua_metop-a_ges_mem018 diag_conv_ges_mem012 fort.222 obs_input.0023
diag_amsua_metop-a_ges_mem019 diag_conv_ges_mem013 fort.223 obs_input.0024
diag_amsua_n15_ges.ensmean diag_conv_ges_mem014 fort.224 obs_input.0025
diag_amsua_n15_ges_mem001 diag_conv_ges_mem015 fort.225 obs_input.0026
diag_amsua_n15_ges_mem002 diag_conv_ges_mem016 fort.226 obs_input.0027
diag_amsua_n15_ges_mem003 diag_conv_ges_mem017 fort.227 obs_input.0028
diag_amsua_n15_ges_mem004 diag_conv_ges_mem018 fort.228 obs_input.0029
diag_amsua_n15_ges_mem005 diag_conv_ges_mem019 fort.229 sigf03
```
This case was a regional analysis with WRF/ARW NetCDF backgrounds. In this case, 20 ensemble members are used to generate the diag files. A brief introduction of the additional files in the GSI observer runs is given below:

- `diag_conv_ges.mem???` - Diag files of conventional observations for ensemble member ???.
- `diag_conv_ges.ensmean` - Diag files of satellite radiance observation for ensemble mean.
- `stdout_mem???` - Standard output from GSI observer run for ensemble member ???.
- `list_run_directory_mem???` - The list of the files in the run directory after the GSI observer is finished for ensemble member ???.

3.3.2 Files in the EnKF Run Directory

After customizing the EnKF run script to your personal environment, it may be submitted to the batch system just as any other job. Upon successful assimilation, the ensemble analyses (both members and ensemble mean), covariance inflation factor to the ensemble analyses, and updated satellite bias correction coefficients (if requested) are output in the run directory. Below is an example of the files generated in the run directory from one of the EnKF regional test cases:
This case was a regional analysis with WRF/ARW NetCDF backgrounds. In this case, 20 ensemble members are used to estimate ensemble covariance, and both conventional observations (prepbufr) and radiance observations (AMSU-A) are assimilated.

A brief introduction of the files is given in the table below:

<table>
<thead>
<tr>
<th>Description</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>stdout</td>
<td>A text output file. This is the most commonly used file to check the analysis processes as well as basic and important information about the analyses. The contents of stdout are explained in detail in Chapter 4 and users are encouraged to read this file to become familiar with the order of EnKF analysis processing.</td>
</tr>
<tr>
<td>firstguess. mem001-0??</td>
<td>ensemble members of WRF background, in NetCDF format. This is in the same format as the WRF forecast. The ensemble background of the analysis variables are extracted from the files.</td>
</tr>
<tr>
<td>firstguess.ensmean</td>
<td>ensemble mean of WRF background, in NetCDF format.</td>
</tr>
<tr>
<td>analysis. mem001-0??</td>
<td>ensemble analysis if EnKF completes successfully. The format is the same as the background file.</td>
</tr>
<tr>
<td>analysis.ensmean</td>
<td>ensemble mean of analyses, in NetCDF format.</td>
</tr>
<tr>
<td>diag_conv_ges. mem001-0??</td>
<td>diag files in binary for conventional and GPS RO observations, which contain the observations and their priors.</td>
</tr>
<tr>
<td>covinflate.dat</td>
<td>Three-dimensional multiplicative inflation factor fields (from function inflate_ens in module inflation). These inflation fields can be visualized using plotting software.</td>
</tr>
<tr>
<td>*info (convinfo, satinfo,...)</td>
<td>info files that control data usage. Please see GSI User guide Chapter 4 for details.</td>
</tr>
<tr>
<td>satbias_in</td>
<td>the input coefficients of bias correction for satellite radiance observations.</td>
</tr>
<tr>
<td>satbias_pc</td>
<td>the input coefficients of satellite radiance bias correction for passive channels.</td>
</tr>
</tbody>
</table>
EnKF Diagnostics and Tuning

This chapter will discuss how to assess whether an EnKF was successful based on the contents of the standard output (stdout). Properly checking the EnKF output will also provide useful information to diagnose potential errors in the system. The chapter begins with an introduction to the content and structure of the EnKF stdout, followed by detailed discussion of tuning options in the namelist. This chapter follows a case at 00z on February 13th, 2014 (case 2014021300). This case uses WRF-ARW NetCDF ensemble files as the background and analyzes several observations typical for operations, including most conventional observation data and select radiance data (AMSU-A, HIRS4). The case was run on a Linux cluster supercomputer, using 32 cores. Users can follow this test to reproduce the following results by visiting:

http://www.dtcenter.org/EnKF/users/tutorial/index.php

4.1 Understanding the Standard Output from EnKF (stdout)

Upon completion of an EnKF run, it is always useful to do a quick check of the standard output (stdout), to assess the performance of the EnKF. The stdout file has information about whether the EnKF analysis has successfully completed, if the ensemble spread inflation looks good, and if the background and analysis fields are reasonable. Understanding the contents of this file can also be very helpful for users to find clues in the event of a crash.

The EnKF stdout has following information:

a. namelist configuration
b. background ensemble members of observations and analysis variables
4. EnKF Diagnostics and Tuning

c. statistics of the ensemble prior
d. domain and observation partition
e. statistics of the ensemble analysis
f. spread inflation of the analysis ensemble

The following section contains a detailed description of the content of stdout, explained using the WRF-ARW case: 2014021300. The analysis domain consists of 129 x 70 x 50 grid points. To keep the output concise and make it more readable, redundant content are omitted.

The following indicates the start of the EnKF analysis. It shows how many processors are used to run the analysis and the beginning time of this run:

Execute poe command line: poe ./enkf.x
running on 32 processors ...

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PROGRAM ENKF_ANL HAS BEGUN. COMPILED 2011319.55 ORG: NP25
STARTING DATE-TIME JUL 14,2016 00:01:03.956 196 THU 2457584

The following lines show how many satellite observation types are set to be read in and if the radiance bias correction is updated before the analysis process:

number of satellite radiance files used 59
number of satellite ozone files used 7

The following lines show the namelist used in the analysis:

namelist parameters:
---------------------
&NAM_ENKF
DATEIN = 2014021300,
DATAPATH = ./,
IASSIM_ORDER = 0,
COVINFLATEMAX = 100.0000 ,
COVINFLATEMIN = 1.000000 ,
DETERMINISTIC = T, SORTINC = T,
CORRLENGTHNH = 500.0000 ,
CORRLENGTHTR = 500.0000 ,
CORRLENGTHSH = 500.0000 ,
... ...
SAVE_INFLATION = F,
LETKF_FLAG = F,
MASSBAL_ADJUST = F,
USE_EDGES = F,
EMISS_BC = T,
ISEED_PERTURBED_OBS = 0
/---------------------

The following lines show ensemble size, background information, the number of 3D analysis variables, and total number of 2D fields of the 3D variables, plus surface pressure (Ps).

20 members
number of background forecast times to be updated = 1
first-guess forecast hours for analysis = 06
5 3d vars to update
total of 251 2d grids will be updated (including ps)
4. EnKF Diagnostics and Tuning

The actual analysis variables and the background type are printed, as well as the maximum and minimum of the surface pressure as a sanity check:

```
Updating U, V, T, QVAPOR, PH, and MU for WRF-ARW...
Surface pressure (spressmn) min/max range: 678.673339843750 1032.22473144531
```

Next, the contents of convinfo are displayed:

```
READ_CONVINFO: tcp 112 0 1 3.000 0 0 0 75.00 0 0 0 0.000 0 0.000 0.000 0.000 2
READ_CONVINFO: ps 120 0 1 3.000 0 0 0 4.000 3.000 1.000 4.000 0.3000E-03 0 0.000 0.000 0.000 0.000 2
  line ignored in convinfo due to use flag ps 132 0
......
  line ignored in convinfo due to use flag gps 44 0
```

The next 272 lines show the content of ozinfo:

```
OZINFO_READ: jpch_oz= 272
1 sbuv6_n14 lev = 1 use = -1 pob = 0.240 gross = 1.000 error = 1.000 b_oz = 10.000 pg_oz = 0.000
2 sbuv6_n14 lev = 2 use = -1 pob = 0.490 gross = 1.000 error = 1.000 b_oz = 10.000 pg_oz = 0.000
3 sbuv6_n14 lev = 3 use = -1 pob = 0.980 gross = 1.000 error = 1.000 b_oz = 10.000 pg_oz = 0.000
   ... ... ...
271 mls30_aura lev = 54 use = -1 pob = 999.999 gross = 9.999 error = 9.999 b_oz = 10.000 pg_oz = 0.000
272 mls30_aura lev = 55 use = -1 pob = 999.999 gross = 9.999 error = 9.999 b_oz = 10.000 pg_oz = 0.000
```

The next 2723 lines show the content of satinfo and starts reading in the radiance bias correction coefficients:

```
RADINFO_READ: jpch_rad= 2723
1 amsua_n15 chan1 var= 3.000 varch_cld= 20.000 use= 1 ermax= 4.500 b_rad= 10.00 pg_rad= 0.00 icld_det=-2
2 amsua_n15 chan2 var= 2.000 varch_cld= 18.000 use= 1 ermax= 4.500 b_rad= 10.00 pg_rad= 0.00 icld_det=-2
3 amsua_n15 chan3 var= 2.000 varch_cld= 12.000 use= 1 ermax= 4.500 b_rad= 10.00 pg_rad= 0.00 icld_det=-2
4 amsua_n15 chan4 var= 0.600 varch_cld= 3.000 use= 1 ermax= 2.500 b_rad= 10.00 pg_rad= 0.00 icld_det=-2
   ... ... ...
2722 ahi_himawari8 chan16 var= 2.200 varch_cld= 0.000 use= -1 ermax= 2.000 b_rad= 10.00 pg_rad= 0.00 icld_det=-2
2723 ahi_himawari8 chan16 var= 2.200 varch_cld= 0.000 use= -1 ermax= 2.000 b_rad= 10.00 pg_rad= 0.00 icld_det=-2
```

The majority of next near 3000 lines show the content of radiance bias correction coefficients:

```
RADINFO_READ: guess air mass bias correction coefficients below
1 amsua_n15 -1.96518 0.00000 39.06817 -0.92319 0.40846 0.00000 0.00173 3.68970 2.36747 8.86294 -1.84829
2 amsua_n15 -7.41540 0.00000 80.13801 -0.81847 -1.01321 0.00000 0.01427 2.63880 8.31509 22.04145 -1.56477
   ... ... ...
2679 avhrr3_metop-b 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
2680 avhrr3_metop-b 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
```

Among the lines describing the radiance bias correction coefficients, the EnKF also starts to inventory the observation number and types for both conventional and radiance data. Observations of various types are read in (from the diag** files) and the number of the observations and time spent reading in the observations are shown:

Start to check the radiance and ozone observations:
4. EnKF Diagnostics and Tuning

Start to check the conventional observations:

<table>
<thead>
<tr>
<th>obtype</th>
<th>nread</th>
<th>nkeep</th>
<th>num_obs_tot</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>8043</td>
<td></td>
<td>7904</td>
</tr>
<tr>
<td>q</td>
<td>2625</td>
<td>2621</td>
<td></td>
</tr>
<tr>
<td>ps</td>
<td>14133</td>
<td>14133</td>
<td></td>
</tr>
<tr>
<td>uv</td>
<td>21982</td>
<td>21982</td>
<td></td>
</tr>
<tr>
<td>sst</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>gps</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>pw</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>dw</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>srw</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>rw</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>tcp</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

A summary of the total number of conventional (1st number 46640), ozone (2nd number 0), and radiance observations (3rd number 6438) in diag files is given:

```
nobs_conv, nobs_oz, nobs_sat = 46640 0 6438
```

Also, the following normalization factors for the radiance bias predictors could be inside the bias correction coefficient lines:

```
npred  = 12
   1 biasprednorm = 1.000000
   2 biasprednorm = 1.000000
   ...
   11 biasprednorm = 1.000000
   12 biasprednorm = 1.000000
```

Then, the time spent reading in the observations and the total number of observations kept and rejected are shown:
### 4. EnKF Diagnostics and Tuning

max time in mpireadobs = 0.6105460

<table>
<thead>
<tr>
<th>total number of obs</th>
<th>53078</th>
</tr>
</thead>
<tbody>
<tr>
<td>53078 obs kept</td>
<td></td>
</tr>
<tr>
<td>0 total obs rejected</td>
<td></td>
</tr>
</tbody>
</table>

time in read_obs = 1.25072427513078 on proc 0

In the stdout, regional averaged (northern hemisphere - NH, southern hemisphere - SH, and tropics - TR) statistics of the ensemble 'priors' to all observations are provided, partly for checking if the inflation is appropriate. The details of this part are discussed in next Section 4.2.

#### innovation statistics for prior:

<table>
<thead>
<tr>
<th>region, obtype, nobs, bias, innov stdev, sqrt(S+R), sqrt(S), sqrt(R):</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH all ps 14083 -0.591E-02 0.860E+00 0.814E+00 0.601E+00 0.550E+00</td>
</tr>
<tr>
<td>TR all ps 50 -0.629E-01 0.422E+00 0.661E+00 0.367E+00 0.550E+00</td>
</tr>
<tr>
<td>NH all t 7766 -0.228E+00 0.163E+01 0.128E+01 0.480E+00 0.119E+01</td>
</tr>
<tr>
<td>TR all t 138 -0.118E+01 0.119E+01 0.234E+00 0.117E+01</td>
</tr>
<tr>
<td>NH all uv 21680 0.136E-01 0.295E+01 0.276E+01 0.140E+01 0.237E+01</td>
</tr>
<tr>
<td>TR all uv 302 -0.379E+00 0.251E+01 0.268E+01 0.124E+01 0.238E+01</td>
</tr>
<tr>
<td>NH all q 2553 -0.323E-01 0.146E+00 0.157E+00 0.868E-01 0.131E+00</td>
</tr>
<tr>
<td>TR all q 68 -0.721E-01 0.202E+00 0.164E+00 0.582E-01 0.154E+00</td>
</tr>
</tbody>
</table>

#### satellite brightness temp

<table>
<thead>
<tr>
<th>instrument, channel #, nobs, bias, innov stdev, sqrt(S+R), sqrt(S), sqrt(R):</th>
</tr>
</thead>
<tbody>
<tr>
<td>anmsua_n18 1 241 -0.199E+00 0.194E+01 0.306E+01 0.176E+01 0.250E+01</td>
</tr>
<tr>
<td>anmsua_n18 2 248 -0.194E+00 0.145E+01 0.234E+01 0.806E+00 0.220E+01</td>
</tr>
<tr>
<td>anmsua_n18 3 248 0.620E+00 0.119E+01 0.206E+01 0.497E+00 0.200E+01</td>
</tr>
<tr>
<td>anmsua_n18 4 248 -0.189E+00 0.398E+00 0.568E+00 0.176E+01 0.250E+01</td>
</tr>
<tr>
<td>anmsua_n18 5 248 0.188E-01 0.337E+00 0.315E+00 0.966E-01 0.300E+00</td>
</tr>
<tr>
<td>anmsua_n18 6 1203 -0.215E+00 0.324E+00 0.206E+01 0.497E+00 0.200E+01</td>
</tr>
<tr>
<td>anmsua_n18 7 1225 -0.396E-01 0.198E+00 0.238E+00 0.140E+01 0.250E+01</td>
</tr>
<tr>
<td>anmsua_n18 8 1155 -0.396E-01 0.198E+00 0.238E+00 0.140E+01 0.250E+01</td>
</tr>
<tr>
<td>anmsua_n18 9 1211 0.408E+00 0.491E+00 0.353E+00 0.429E+00 0.350E+01</td>
</tr>
<tr>
<td>anmsua_n18 10 1211 0.720E+00 0.723E+00 0.404E+00 0.567E+01 0.400E+00</td>
</tr>
<tr>
<td>anmsua_n18 11 12 0.311E+00 0.198E+00 0.380E+01 0.148E+01 0.350E+01</td>
</tr>
<tr>
<td>hirs4_metop-a 3 1 0.115E+01 0.115E+01 0.534E+00 0.631E+00 0.530E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 4 42 0.260E+00 0.521E+00 0.403E+00 0.488E+00 0.400E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 5 7 -0.534E+00 0.558E+00 0.367E+00 0.696E+00 0.360E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 6 7 -0.748E+00 0.782E+00 0.473E+00 0.109E+00 0.460E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 7 9 -0.557E+00 0.815E+00 0.580E+00 0.108E+00 0.570E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 8 9 0.374E+00 0.675E+00 0.100E+00 0.208E+01 0.100E+01</td>
</tr>
<tr>
<td>hirs4_metop-a 9 10 0.199E+00 0.581E+00 0.612E+00 0.120E+00 0.600E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 10 11 9 -0.305E+01 0.718E+00 0.143E+01 0.120E+01</td>
</tr>
<tr>
<td>hirs4_metop-a 11 12 3 0.988E+00 0.179E+01 0.213E+01 0.141E+01 0.160E+01</td>
</tr>
<tr>
<td>hirs4_metop-a 12 13 9 0.694E-01 0.330E+00 0.382E+00 0.116E+00 0.364E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 13 14 9 0.369E-01 0.257E+00 0.294E+00 0.137E+00 0.260E+00</td>
</tr>
<tr>
<td>hirs4_metop-a 14 15 9 0.352E+01 0.289E+00 0.281E+00 0.107E+00 0.260E+00</td>
</tr>
</tbody>
</table>

Next, the analysis variable fields and the observations are distributed to the different processors. The following lines show the maximum and minimum number of observations and grid on subdomain and the time used to setup those decompositions:

<table>
<thead>
<tr>
<th>time in estimate_work_enkf1 = 5.723375361412764E-002 secs</th>
</tr>
</thead>
<tbody>
<tr>
<td>min/max numbobs</td>
</tr>
<tr>
<td>min/max estimated work</td>
</tr>
<tr>
<td>npts =</td>
</tr>
<tr>
<td>min/max number of points per proc = 243</td>
</tr>
<tr>
<td>time to do model space decomp = 1.529277069494128E-003</td>
</tr>
<tr>
<td>nobtot = 53078</td>
</tr>
<tr>
<td>min/max number of obs per proc = 1658 1659</td>
</tr>
<tr>
<td>time to do ob space decomp = 3.935033455491066E-004</td>
</tr>
<tr>
<td>sending out observation prior ensemble perts from root ...</td>
</tr>
<tr>
<td>nobtot*nanals</td>
</tr>
<tr>
<td>npts*ndim</td>
</tr>
<tr>
<td>... took 1.730861375108361E-002 secs</td>
</tr>
<tr>
<td>time in load_balance = 7.92941360399196E-002 on proc 0</td>
</tr>
</tbody>
</table>
4. EnKF Diagnostics and Tuning

Then, the ensemble background members of the analysis variables are read in and the maximum and minimum values of the fields at each vertical level are displayed. The maximum and minimum values are useful for a quick confirmation that the background fields have been read successfully. The size of the real array of ensemble perturbations updated on each processor and the time spent to read in and distribute the background ensemble are also shown:

```
anal_chunk size = 1591340
READGRIDDATA_ARW: U 1 -20.55222 20.72632
... READGRIDDATA_ARW: U 50 -12.60845 52.03727
READGRIDDATA_ARW: V 51 -11.67894 21.03603
... READGRIDDATA_ARW: V ...
READGRIDDATA_ARW: V 100 -17.20951 21.87867
READGRIDDATA_ARW: T 101 -62.00526 13.33912
... READGRIDDATA_ARW: T 150 469.5381 509.8757
READGRIDDATA_ARW: QVAPOR 151 8.417385E-05 1.721506E-02 ...
READGRIDDATA_ARW: QVAPOR 200 1.5460877E-06 1.0506745E-05
READGRIDDATA_ARW: PH 201 -14.64355 4.688477 ...
READGRIDDATA_ARW: PH 250 7195.078 14206.67
READGRIDDATA_ARW: MU 251 -1326.610 3290.569
time in readgridata on root 0.326851664343849 secs

time to scatter state on root 0.163762195268646 secs

time in read_ensemble = 0.494254242163152 on proc 0
```

In EnSRF, observations can be skipped/not assimilated due to the adaptive observation thinning. The following lines show how many observations are skipped or not assimilated by this thinning.

```
assimilate obs in order they were read in
1 timing on proc 0 = 0.51 0.24 0.01 0.01 0.00 0.10 0.00 0.15 0
22152 out of 53078 obs skipped, 30926 used
22736 out of 30926 same lat/long

time to broadcast obfit_post = 5.432746838778257E-004 secs, niter = 1
```

Next, the update to the analysis variables is performed and the time for the updating is shown:

```
time in enkf_update = 4.79503449611366 on proc 0
```

The regional averaged statistics of the inflation values are shown for surface pressure:

```

global ps prior std. dev min/max = 21.52301 172.3270
NH mean ps prior standard deviation = 87.82042
NH mean ps posterior standard deviation (before inflation)= 29.20955
NH mean ps posterior standard deviation (after inflation)= 54.58403
NH mean ps inflation = 2.421974
TR mean ps prior standard deviation = 31.09247
TR mean ps posterior standard deviation (before inflation)= 28.31872
TR mean ps posterior standard deviation (after inflation)= 30.79383
TR mean ps inflation = 1.106122

time in inflate_ens = 1.366839744150639E-002 on proc 0
```

After the EnKLF analysis, innovation statistics of the ensemble analyses are shown, similar to the ensemble prior's fit to all observations:
4. EnKF Diagnostics and Tuning

innovation statistics for posterior:

| region, obtype, nobs, bias, innov stdev, sqrt(S+R), sqrt(S), sqrt(R) |
|---------------------------------|------------------|------------------|------------------|------------------|
| NH all ps                      | 14083            | 0.503E-03        | 0.757E+00        | 0.569E+00        | 0.146E+00        | 0.550E+00        |
| TR all ps                      | 50               | -0.562E-01       | 0.398E+00        | 0.585E+00        | 0.263E+00        | 0.519E+00        |
| NH all t                       | 138              | -0.107E+01       | 0.118E+01        | 0.199E+00        | 0.117E+01        |
| NH all uv                      | 21850            | 0.406E-01        | 0.267E+01        | 0.246E+00        | 0.655E+00        | 0.237E+01        |
| TR all uv                      | 302              | -0.259E+00       | 0.236E+01        | 0.246E+00        | 0.636E+00        | 0.238E+01        |
| NH all q                       | 5533             | -0.174E-01       | 0.114E+01        | 0.138E+00        | 0.444E-01        | 0.131E+00        |
| TR all q                       | 68               | -0.279E-01       | 0.168E+00        | 0.159E+00        | 0.413E-01        | 0.154E+00        |

satellite brightness temp

| instrument, channel #, nobs, bias, innov stdev, sqrt(S+R), sqrt(S), sqrt(R) |
|---------------------------------|------------------|------------------|------------------|------------------|
| amsua_n18 1                     | 241              | 0.614E+02        | 0.915E+02        | 0.263E+01        | 0.823E+00        |
| amsua_n18 2                     | 248              | -0.132E+03       | 0.171E+03        | 0.223E+01        | 0.376E+00        |
| amsua_n18 3                     | 248              | -0.303E+02       | 0.398E+00        | 0.201E+01        | 0.244E+00        |
| amsua_n18 4                     | 248              | -0.231E+01       | 0.580E+01        | 0.555E+00        | 0.776E-01        |
| amsua_n18 5                     | 248              | -0.788E+00       | 0.199E+01        | 0.305E+00        | 0.575E-01        |
| amsua_n18 6                     | 1203             | -0.138E+01       | 0.158E+01        | 0.234E+00        | 0.423E-01        |
| amsua_n18 7                     | 1225             | -0.273E+01       | 0.287E+01        | 0.232E-01        |
| amsua_n18 8                     | 1155             | -0.131E+01       | 0.186E+01        | 0.254E+00        | 0.437E-01        |
| amsua_n18 10                    | 1211             | -0.179E+01       | 0.196E+01        | 0.352E-01        |
| amsua_n18 11                    | 12               | -0.130E+01       | 0.133E+01        | 0.404E-01        |
| amsua_n18 15                    | 245              | -0.163E+03       | 0.198E+03        | 0.357E+01        |
| hirs4_metop-a 3                 | 1                | 0.118E+01        | 0.118E+01        | 0.533E+00        |
| hirs4_metop-a 4                 | 42               | 0.286E+00        | 0.542E+00        | 0.403E+00        |
| hirs4_metop-a 5                 | 7                | -0.451E+00       | 0.487E+00        | 0.364E+00        |
| hirs4_metop-a 6                 | 7                | -0.724E+00       | 0.755E+00        | 0.466E+00        |
| hirs4_metop-a 7                 | 9                | -0.572E+00       | 0.817E+00        | 0.577E-01        |
| hirs4_metop-a 8                 | 9                | 0.122E+01        | 0.136E+01        | 0.100E+01        |
| hirs4_metop-a 10                | 9                | 0.855E+00        | 0.100E+01        | 0.104E+00        |
| hirs4_metop-a 11                | 9                | -0.236E+01       | 0.543E+00        | 0.130E+01        |
| hirs4_metop-a 12                | 34               | -0.738E+00       | 0.153E+01        | 0.178E+01        |
| hirs4_metop-a 13                | 9                | 0.121E+01        | 0.128E+01        | 0.375E-01        |
| hirs4_metop-a 14                | 9                | 0.776E-01        | 0.256E+00        | 0.281E-01        |
| hirs4_metop-a 15                | 9                | 0.281E+00        | 0.384E+00        | 0.273E+00        |

Finally, the minimum and maximum of the analysis increments are shown as below. The analysis increments should be within a reasonable range. The computational time of these steps are also shown:

- time to gather state on root: 0.101591360988095 secs
- time level 1
- time ens mean increment min/max ps: -123.0895 264.4738
- time ens mean increment min/max var: -7.882635 7.015289
- time ens mean increment min/max var: -7.460623 6.953684
- time in writegriddata on root: 0.255841280566528 secs
- time in write_ensemble = 0.385106393136084 on proc 0

The next line indicates the time of the analysis finish:

ENDING DATE-TIME JUL 14, 2016 00:01:11.329 196 THU

All done.

The next lines show the computer resources used in the analysis and a sign that EnKF has finished, "all done".
4. EnKF Diagnostics and Tuning

***************RESOURCE STATISTICS**********************
The total amount of wall time = 7.431095
The total amount of time in user mode = 6.480014
The total amount of time in sys mode = 0.713891
The maximum resident set size (KB) = 122848
Number of page faults without I/O activity = 46688
Number of page faults with I/O activity = 0
Number of times filesystem performed INPUT = 0
Number of times filesystem performed OUTPUT = 0
Number of Voluntary Context Switches = 24121
Number of InVoluntary Context Switches = 138
****************************************************************

all done!

4.2 Tuning of Inflation and Localization

Proper inflation and localization values need to be determined by experimenting with different inflation values. The goal is to make the total ensemble spreads of priors match the innovations as much as possible. In the tuning process, the vertical and horizontal structure of the match should be carefully examined. The 3D distribution of the inflation values can be plotted offline from the output file "covinflate.dat".

The tuning processes can vary for different resolutions and physics of the models as well as the observation types assimilated. The performance of the tuning may be better examined after multiple assimilation cycles allowing their effects to be accumulated and converged in cycling assimilation. Please also note that proper settings of observation error estimations are also critical for the inflation tuning.

Next, the test case is taken as an example to check if the inflation is appropriate. In the following lines of the standard output (stdout), the number (column nobs), mean (column bias), and standard deviation (column innov stdev) of the ensemble background to Ps, wind, temperature, and water vapor observations in the NH and TR are shown:

innovation statistics for prior: conventional obs
region, obtype, nobs, bias, innov stdev, sqrt(S+R), sqrt(S), sqrt(R):
NH all ps 14083 -0.591E-02 0.860E+00 0.814E+00 0.601E+00 0.550E+00
TR all ps 50 -0.629E-01 0.422E+00 0.661E+00 0.367E+00 0.550E+00
NH all t 7766 -0.228E+00 0.163E+01 0.128E+01 0.480E+00 0.119E+01
TR all t 138 -0.130E+00 0.118E+01 0.119E+01 0.234E+00 0.117E+01
NH all uv 21680 0.136E-01 0.295E+01 0.276E+01 0.140E+01 0.237E+01
TR all uv 302 -0.379E+00 0.251E+01 0.268E+01 0.124E+01 0.238E+01
NH all q 2553 -0.323E-01 0.146E+00 0.157E+00 0.868E-01 0.131E+00
TR all q 68 -0.721E-01 0.202E+00 0.164E+00 0.582E-01 0.154E+00

It would be desirable that the total ensemble spreads of prior/background match the innovation standard deviations as close as possible (Equation 15). In practice, one should check if there is any substantial ensemble spread deficiency. In the above sample standard output lines, the ensemble prior spreads are shown in the column $\sqrt{S}$, and the total spreads in the column $\sqrt{S + R}$. Here, $\sqrt{R}$ is the observation error.

The above sample statistics shows that in NH, the total spread of temperature observation priors (0.128E+01) is close to the innovation standard deviation (0.163E+01) of the priors fit to the observations.
4. Tuning EnKF through Key Namelist Options

The namelist parameters controlling the EnSRF analysis are set in the namelist sections: /enkf_nml/, /satobs_enkf/, and /nam_wrf/ (all in the file enkf.nml). To obtain a successful analysis, it is very important to setup/tune properly the namelist variables, particularly those related to inflation and localization, for each specific application configuration of various models and observation types.

4.3.1 Set Up the Analyses Time and Data Location

The analysis time and working location are setup by the following parameters:

- **datein**: analysis time (YYYYMMDDHH)
- **datapath**: path to data directory (include trailing slash). In most cases, this should point to the current directory because the run scripts have setup the run environment in the working directory including copying the EnKF executable and data into the working directory.

4.3.2 Set Up Analysis Algorithm

In the current implementation, a few variations/flavors of the EnKF are available, including the ensemble square root Kalman filter (EnSRF, Whitaker et al., 2010, etc), the perturbed observations EnKF, and the local transformed Kalman filter (LETKF). These options are set in the namelists below:

- **deterministic**
  - = true, use EnSRF without perturbed observations;
  - = false, use perturbed observations version of EnKF.
- **sortinc**
  - if deterministic = false, re-order observations to minimize regression errors as described in Anderson (2003).
- **letkf_flag**
  - if true, use the LETKF scheme.
- **boxsize**
  - observation box size for LETKF (deg)
  - = 0, assimilation of observations in the order they are read in;
  - = 1 in random order;
  - = 2 in order of predicted posterior variance reduction.
4. EnKF Diagnostics and Tuning

\textit{paoverp\_thresh} the threshold of the ratio of observation space posterior variance divided by prior variance ($\leq 1.0$), that is, the expected reduction of prior variance by the observations.

Smaller values mean only those observations with stronger impact will be assimilated.

If $=1$, no thinning of observations is done. Typical values: 0.99 or 0.98.

4.3.3 Set Up the Analyses Variables

The analyses variables for both global and regional models are hardcoded in the EnSRF. There are several options using the following parameters:

\begin{itemize}
  \item \textit{nvars}: number of 3d model variables to update.
\end{itemize}

For hydrostatic global models, there are typically 5, including U, V, T, QVAPOR, Ozone. For non-hydrostatic regional models, like WRF, different combinations of the 3D and 2D analyses variables are set as below:

\begin{itemize}
  \item \textbf{For ARW} (\textit{arw =true})
    \begin{itemize}
      \item \textit{nvars} =3: U, V, T, and MU
      \item \textit{nvars} =4: U, V, T, QVAPOR, and MU
      \item \textit{nvars} =5: U, V, T, QVAPOR, PH, and MU
      \item \textit{nvars} =6: U, V, W, T, QVAPOR, PH, and MU
    \end{itemize}
  \item \textbf{For NMM} (\textit{nmm =true})
    \begin{itemize}
      \item \textit{nvars} =3: U, V, T, and PD
      \item \textit{nvars} =4: U, V, T, QVAPOR, and PD
      \item \textit{nvars} =5: U, V, T, QVAPOR, CWM, and PD
    \end{itemize}
\end{itemize}

The analyses variable list can be adjusted or additional variables, such as moisture related variables, can be added to the list by modifying the subroutines "gridinfo.F90" and "gridio.F90" accordingly.

\begin{itemize}
  \item \textit{pseudo\_rh}: use or not 'pseudo-rh' analysis variable, as in GSI.
  \item \textit{cliptracers}: if true, tracers are clipped to zero when read in, and just before they are written out.
\end{itemize}

4.3.4 Set Up the Ensemble Backgrounds

The following parameters define which background fields will be used in the analyses and their dimensions:
4. EnKF Diagnostics and Tuning

regional = true, perform regional EnSRF analyses using either ARW or NMM inputs as the background.
            =false, perform a global analysis. If either the parameter nmm or wrf is set to true, it will be set to true.

nmm if true, background comes from WRF NMM. When using other background fields, set it to false.

wrf if true, background comes from WRF ARW. When using other background fields, set it to false.

nlons number of grid points in longitude of the model background

nlats number of grid points in latitude of the model background

nlevs number of vertical levels of the model background

nanls number of ensemble members

4.3.5 Set Up Localization Distances

In the current EnKF implementation, distances for localization can be set separately in the northern hemisphere, tropics and southern hemisphere, and in the horizontal, vertical and time dimensions, and for different observation types using namelist parameters. The length scales should be given in km for the horizontal, hours for time, and scale heights (units of $-\log(P/P_{ref})$) in the vertical.

There are two options to setup the localization distances in horizontal and vertical. These are decided by the namelist variable "readin_localization":

readin_localization: =.true., customized horizontal and vertical localization values varying with model levels are read in from the external text file "hybens_locinfo".
            =.false., the horizontal and vertical localization distances are set by the following parameters

  corrlengthnh: length for horizontal localization in km in the northern hemisphere (25N-90N, NH)
  corrlengthtr: length for horizontal localization in km in the tropics (25S-25N, TR)
  corrlengthsh: length for horizontal localization in km in the southern hemisphere (25S-90S, SH)
  lnsigcuto: scale height for vertical localization in $-\log(P/P_{ref})$ in NH.
  lnsigcutofftr: scale height for vertical localization in $-\log(P/P_{ref})$ in TR.
  lnsigcutoffsh: scale height for vertical localization in $-\log(P/P_{ref})$ in SH.

The text file "hybens_locinfo" contains vertical profile of horizontal and vertical localization length scales (in e-folding scale). These scales are converted to the full distance width ($^*1/0.388$ in km) of Gaspari-Cohn function, where it goes to zero. The horizontal and vertical scales can be adjusted for each observation types easily in the subroutine "read_locinfo.f90".

For satellite radiance and surface pressure observations, the vertical localization distances are set separately using the following parameters:
4. EnKF Diagnostics and Tuning

\texttt{lnsigcutoffsatnh, lnsigcutoffsattr, lnsigcutoffsatsh}; and \texttt{lnsigcutoffpsnh, lnsigcutofffstr, lnsigcutofffssh}

There is also one option to setup the time localization window, which is the time away from the analyses time. This is decided by the following namelist variables:

\begin{itemize}
  \item \texttt{obtimelnh}: time in hours away from the analyses time to move 2800 km at 30 ms$^{-1}$ in the northern hemisphere
  \item \texttt{obtimelttr}: same as \texttt{obtimelnh} but for the tropics
  \item \texttt{obtimelsh}: same as \texttt{obtimelnh} but for the southern hemisphere
\end{itemize}

The empirical determination/tuning of proper localization distances is important for successful analyses with ensemble data assimilation. In general, large horizontal scale observations, like GPS radio occultation and radiosonde observations, require larger horizontal localization distances to fully take advantage of the observations. On the other hand, high horizontal resolution observations, like radar observations, may need shorter horizontal localization distances. The same is true for the localization distance in vertical.

4.3.6 Set Up Adaptive Posterior Inflation Parameters

The inflation can be set up by the following parameters:

\begin{itemize}
  \item \texttt{anapertwtnh}: inflation parameter in NH.
  \item \texttt{anapertwtr}: inflation parameter in TR.
  \item \texttt{anapertwsh}: inflation parameter in SH.
\end{itemize}

The parameters $= 0$ means no inflation.
$= 1$ means inflation all the way back to prior spread.

These are the key tuning parameters for the performance of EnSRF. Typical values: $0.95$.

The inflation factor fields can be smoothed out using the following parameter:

\begin{itemize}
  \item \texttt{smoothparm}: parameter for smoothing inflation factor,
  \hspace{1cm} $=-1$ for no smoothing.
  \hspace{1cm} $>0$, the estimated inflation factor is smoothed using a Gaussian spectral filter with an e-folding scale of the parameter
  \item \texttt{latbound}: where the transition latitude starts ($=25N$ or $25S$)
  \item \texttt{delat}: latitude width of transition zone where the inflation parameter is smoothed.
\end{itemize}

The minimum and maximum inflation values allowed can be controlled by the following parameters:

\begin{itemize}
  \item \texttt{covinflatemin}: minimum inflation factor
  \item \texttt{covinflatemax}: maximum inflation factor
\end{itemize}
4. EnKF Diagnostics and Tuning

4.3.7 Satellite Observations Related Parameters

The following parameters are used to setup the adaptive satellite bias correction:

- \( \text{lupd_satbias} \): = .true., update the satellite bias correction coefficients using the adaptive bias correction scheme (when the iterations of numiter > 1).
  
  = .false., the bias correction coefficients are not updated.

- \( \text{numiter} \): number of iterations for state/bias correction coefficients update. (only relevant when satellite radiances assimilated, i.e. nobs_sat>0)

- \( \text{biasvar} \): background error variance for radiance bias coefficients (used in radbias.f90). The default is GSI value.

- \( \text{saterrfact} \): factor to multiply satellite radiance errors.

- \( \text{sattypes_rad} \): list of satellite observation types to be assimilated.

- \( \text{sattypes_oz} \): list of ozone observation types to be assimilated.

4.3.8 Observation QC Parameters

In addition to the observation quality flags decided in the GSI observer, the EnSRF itself also conducts a gross check on the observations.

The outlier test computes the difference between the observation value and the prior ensemble mean. It then computes a standard deviation by taking the square root of the sum of the observation error variance and the prior ensemble variance for the observation. If the difference between the ensemble mean and the observation value is more than the specified number of standard deviations, then the observation is tossed. The threshold of the check is set as following:

- \( \text{sprd_tol} \): tolerance for gross check of observations.

  Reject a observation if the prior mean is more than this many \( \sqrt{(S + R)} \) from the observation, where \( S \) is ensemble prior variance and \( R \) is observation error variance. Typical value: 3.0

The following parameters are used to control if the GSI quality control procedure is performed in EnSRF:

- \( \text{varqc} \): = .false., no variational quality control performed (default).

- \( \text{huber} \): = .true., use huber norm instead of "flat-tail" (default: .false.).

- \( \text{zhuberleft} \): departure parameter of huber norm.

- \( \text{zhuberright} \): departure parameter of huber norm.
Applications for Regional and Global EnKF

In this chapter, the elements from the previous chapters will be applied to demonstrate how to run a regional and global case using the GSI observer and EnKF. These examples are intended to give users a clear idea of how to set up the GSI observer and EnKF for a particular application and properly check the run status and analysis results in order to determine if the run was successful. Note that the regional example focuses on WRF ARW, however WRF NMM and NMMB runs are similar, but require different background ensemble and namelist options. Similarly, the global example features a single global configuration (T254), however users may wish to use a different configuration, again requiring different background ensemble and namelist options.

It is assumed that the reader has successfully compiled GSI and EnKF on a local machine. For regional case studies, users should have the following data available:

1. Ensemble for background
   - Ensemble files from WRF-ARW, WRF-NMM, NMM-B forecast files may be used. For this case, WRF-ARW ensemble members will be used, which are generated from the GFS ensemble, following the naming convention: wrfarw.mem00nn. Where nn corresponds to each ensemble member ("ensmean" appended for ensemble mean).

2. Conventional, GPSRO and radiance data
   - Real time GDAS and NAM PrepBUFR data can be obtained from the server: ftp://ftpprd.ncep.noaa.gov/pub/data/nccf/com/gfs/prod/
5. Applications for Regional and Global EnKF

ftp://ftpprd.ncep.noaa.gov/pub/data/nccf/com/nam/prod/

3. Fixed files
   • Fixed files are located in the comGSIv3.7_EnKFv1.3/fix directory

For **global case studies**, users should have the following data available:

1. Ensemble for background
   • Ensemble files from GFS forecast files may be used. GFS ensemble members corresponding to various spectral resolutions may be used, following the naming convention: sfg_yyymmddhh_fhrff_mem0nn. The "yyymmddhh" corresponds to the initialization date, "fhr" corresponds to the forecast hour, and "nn" corresponds to each ensemble member (ensmean appended for ensemble mean).

2. Conventional data
   • Real time GDAS PrepBUFR (and BUFR) data can be obtained from the server: ftp://ftpprd.ncep.noaa.gov/pub/data/nccf/com/gfs/prod/

3. Fixed files
   • Fixed files are located in the comGSIv3.7_EnKFv1.3/fix directory

The following cases will give users an example of how to run the EnKF for a regional case with various data sources, as well as a simple global application case. Users are welcome to download these examples from the EnKF User’s webpage (online case for release v1.1) or create a new background ensemble and obtain observation data from the above server.

The background ensemble files and observations used in the regional case studies are as follows:

1. Background files: wrfarw.mem0001 - wrfarw.mem0020, wrfarw.ensmean
   • The horizontal grid spacing is 45-km with 51 vertical sigma levels.

   ![Figure 5.1: Background used for regional case study](image)

2. Conventional, GPSRO and radiance data from 00 UTC 13 February 2014
   • Conventional file:
     – gdas1.t00z.prepbufr.nr
   • GPSRO file:
     – gdas1.t00z.gpsro.tm00.bufr_d
5. Applications for Regional and Global EnKF

- Radiance files:
  - gdas1.t00z.1bamua.tm00.bufr_d
  - gdas1.t00z.1bhrs4.tm00.bufr_d
  - gdas1.t00z.1bmhs.tm00.bufr_d

3. Fixed files:
- Fixed files are located in the GSI/EnKF community code package, under /comGSIv3.7_EnKFv1.3/fix
- For observation control:
  - convinfo: conventional data (PrepBUFR, GPSRO) info file
  - ozinfo: ozone retrieval info file
  - satinfo: satellite radiance info file
- Adaptive radiance bias correction. Although example files are provided in the /fix directory, it is better to obtain bias correction files valid for the proper date. For this case, GFS bias correction coefficients are provided with the case data:
  - comgsi_satbias_in: satellite bias correction coefficient file
  - comgsi_satbias_pc_in: satellite bias correction coefficient file for passive channels.

This case study was run on a Linux cluster. GSI no longer (from GSI v3.2) requires byte-swapping to little endian format. BUFRLIB can automatically handle byte order issues.

For the **regional ARW case**, assume the following locations:

Path to the background ensemble files:

```
/scratch/casedata/enkf_arw_2014021300/bk/
```

the path to the observations:

```
/scratch/casedata/enkf_arw_2014021300/obs
```

and the GSI release version 3.7/EnKF version 1.3 is located at:

```
/scratch/comGSIv3.7_EnKFv1.3
```

For the global GFS case, assume locations as follows: Path to background ensemble files:

```
/scratch/casedata/enkf_glb_t62/bkg
```

the path to the observations:

```
/scratch/casedata/enkf_glb_t62/obs
```

and the GSI release version 3.7/EnKF version 1.3 is located at:

```
/scratch/comGSIv3.7_EnKFv1.3
```
5. Applications for Regional and Global EnKF

5.1 Running GSI Observer for Regional Applications

5.1.1 Run Script

With both GSI and EnKF compiled and the background ensemble files and observations acquired, the next step is to work with the gsi observer run script, comgsi_run_regional.ksh. The location of this script is under comGSIv3.7_EnKFv1.3/ush. This run script is the same as the one used for a GSI analysis run, with a few specific options selected in order to loop through all the ensemble members and generate the ensemble observation priors for each member and the ensemble mean. In addition to the GSI observer specific options, other user-specific modifications need to be made:

- **Set up batch queueing system**
  To run GSI with multiple processors, a job queueing head must be added to the comgsi_run_regional.ksh script. The set up of the job queue is dependent on the machine and the job control system. Refer to the GSI User’s Guide, section 3.2.2, for more examples of the setup section of this script. The following example is setup to run on a Linux cluster with PBS:

  ```
  #PBS -A P????????
  #PBS -l walltime=00:20:00
  #PBS -N observer-allobs
  #PBS -l select=1:ncpus=4:mpiprocs=4
  #PBS -q regular
  #PBS -o out.observer-allobs
  #PBS -j oe
  ```

- **Set up number of processors and the job queue system used** For this example, LINUX_PBS and 4 processors are used:
  ```
  GSIPROC=4
  ARCH='LINUX_PBS'
  ```

- **Set up the case data, analysis time, fix files, GSI executable, and CRTM coefficients:**
  **Set up analysis time:**
  ```
  ANAL_TIME=2014021300
  ```

  Set up working directory, which will hold the analysis results (all ensemble members will be run in this directory). This directory must have the proper write permissions as well as enough space to hold the output.
  ```
  JOB_DIR=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/gsidiag_2014021300
  ```

  Set path to the observation directory and the PrepBUFR file within the observation directory. All observations to be assimilated should reside in this directory.
  ```
  WORK_ROOT=/scratch/${user}/casedata/enkf_arw_2014021300/obs
  PREPBUFR=${OBS_ROOT}/gdas1.t00z.prepbufr.nr
  ```

  Set path to background ensemble files:
  ```
  BK_ROOT=/scratch/${user}/casedata/enkf_arw_2014021300/bkg
  ```

  Set file for background ensemble mean:
Set the GSI system used for this case, including the paths of the fix files and the CRTM coefficients as well as the location of the GSI executable:

- CRTM_ROOT=/scratch/${user}/CRTM_v2.3.0
- GSI_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/
- FIX_ROOT=${GSI_ROOT}/fix
- GSI_EXE=${GSI_ROOT}/build/bin/gsi.exe
- GSI_NAMELIST=${GSI_ROOT}/ush/comgsi_namelist.sh

- Set which background and background error file to use

  - bk_core=ARW
  - bkcv_option=NAM
  - if_clean=clean

  This example uses the ARW NetCDF background; therefore bk_core is set to ARW. The regional background error covariance file is used in this case, as set by bkcv_option=NAM. Finally, the run scripts are set to clean the run directory to delete all temporary intermediate files.

- Choose to run GSI observer, set up background ensemble information

  - if_observer=Yes
  - no_member=20
  - BK_FILE_mem=${BK_ROOT}/wrfarw.mem

  The option if_observer=Yes is the switch that enables run_gsi_regional.ksh to run the GSI observer (rather than GSI analysis). In this example, 20 ensemble members are selected with the naming convention: wrfarw.memnnnn. Note that memnnnn which is associated with each ensemble member (nnnn), is not included and will be appended later in the script.

- Link observations

  - # Link to the prepbufr data
  - ln -s ${PREPBUFR} ./prepbufr
  - # Link to the radiance data
  - ...
  - ln -s ${srcobsfile[$ii]} ${gsiobsfile[$ii]}
  - ...

Past the arch selection, environment variable checks, and creation of working directory, users will find the location where the observations are linked. For this case, the conventional PrepBUFR observations have been linked, as well as three different satellite radiance BUFR files (AMSU-A, HIRS4, and MHS) and a GPS RO BUFR file. These files will be linked to the working directory and observation innovations (diag) files will be generated for each observation.

In the run script, the proper anavinfo file is selected based on the core and background error covariance used for the case:

```bash
echo ' Use NAM background error covariance'
BERROR=${FIX_ROOT}/${BYTE_ORDER}/nam_nmmstat_na.gcv
OBERROR=${FIX_ROOT}/nam_errtable.r3dv
```
5. Applications for Regional and Global EnKF

The anavinfo file used for this case is `anavinfo_arw_netcdf`, because the background is NetCDF ARW and the regional background error covariance has been selected. It is important to make sure the number of vertical levels in `anavinfo_arw_netcdf` must match those for the background. The example run script provided with the release code, `comgsi_run_regional.ksh`, takes care of this automatically for the regional ARW case.

5.1.2 Run GSI Observer and Check Run Status

Once the run script is set up properly for the case data and the machine environment, GSI can be run through the run script. The following command will submit the job, for LINUX PBS:

```
$ qsub comgsi_run_regional.ksh
```

While the job is running, move to the working directory and check the details. Given the following working directory setup:

```
WORK_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/gsidiag_2014021300
```

Go to directory `WORK_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/` and check the run directory. A directory named `gsidiag_2014021300` should have been created. This is the directory for this GSI observer run. The directory will be populated with many files such as:

- `amsua_n18.TauCoeff.bin`
- `ssmi_f15.SpcCoeff.bin`
- `amsua_n18.SpcCoeff.bin`
- `ssmi_f15.TauCoeff.bin`
- `imgr_g11.SpcCoeff.bin`
- `imgr_g11.TauCoeff.bin`

... ...

These files are CRTM coefficients that have been linked to this run directory through the GSI run script. Similar to running the GSI analysis, many other files are linked or copied to this run directory, such as:
5. Applications for Regional and Global EnKF

\texttt{gsiparm.anl}: GSI namelist

\texttt{prepbufr}: PrepBUFR file for conventional observation

\texttt{convinfo}: data usage control for conventional data

\texttt{satbias.in}: satellite bias correction coefficient file

\texttt{satinfo}: data usage and channel control for satellite radiance data

\texttt{berror.stats}: background error file

\texttt{errtable}: observation error file

Additionally, for the GSI observer many files are generated as a result of the \texttt{read.obs.save} and \texttt{read.obs.skip} options in the namelist for writing/reading collective observation selection information:

\texttt{obs_input.nnnn}: During ensemble mean (\texttt{save=True}, \texttt{skip=False}), save all observation preprocessing to this file. Each file (\texttt{nnnn}) is for a different variable type.

\texttt{pennnn.obs_setup}: When looping through ensemble members (\texttt{save=False}, \texttt{skip=True}), all observation preprocessing is skipped, \texttt{obs_input.nnnn} files are read and \texttt{pennnn.obs_setup} is written for each processor (\texttt{nnnn}) for all observations.

As the GSI observer is running for the ensemble mean as well as looping through each member, files are generated for each ensemble member, as well as the ensemble mean.

\texttt{pennnn.conv_01}: Files generated after O-B, generated for conventional observations as well as one file per sensor (e.g. AMSU-A n18). These files are trimmed for resulting diag files and cleaned.

\texttt{list_run_directory}: Directory listing after ensemble mean is run, before directory is cleaned for running the ensemble members.

\texttt{list_run_directory_memnnn}: As above, for each ensemble member (\texttt{nnn}).

\texttt{stdout}: Standard output file for ensemble mean.

\texttt{stdout_memnnn}: Standard output file for each ensemble member (\texttt{nnn}).

The presence of the standard output files in the directory suggest the GSI observer run scripts have successfully set up a run environment for the GSI observer, properly looping through the ensemble members, and the GSI executable is running on each ensemble member.

Once GSI has finished running, diag files should be generated for each observation type for each member as well as the ensemble mean:

\texttt{diag_conv_ges.ensmean} \quad \texttt{diag_conv_ges.mem001}
\texttt{diag_conv_ges.mem002} \quad \texttt{diag_conv_ges.mem003}
\texttt{diag_conv_ges.mem004} \quad \texttt{diag_conv_ges.mem005}
\texttt{...} \quad \texttt{...} \quad \texttt{...} \quad \texttt{...}

\texttt{diag_amsua_metop-a_ges.ensmean} \quad \texttt{diag_amsua_metop-a_ges.mem001}
\texttt{diag_amsua_metop-a_ges.mem002} \quad \texttt{diag_amsua_metop-a_ges.mem003}
\texttt{diag_amsua_metop-a_ges.mem004} \quad \texttt{diag_amsua_metop-a_ges.mem005}
\texttt{...} \quad \texttt{...} \quad \texttt{...} \quad \texttt{...}
5. Applications for Regional and Global EnKF

5.1.3 Check for Successful GSI Completion

The presence of the diag files and standard out files for the ensemble mean and each member indicates that the GSI observer has run without crashing, but does not necessarily indicate a successful analysis. It is important to check the stdout files in the run directory to make sure the GSI observer completed each step without any obvious problems. The following are several important areas of the standard out file to check:

1. Read in the anavinfo and namelist

```
gsi_metguess_mod*init_: 2D-MET STATE VARIABLES:
ps
z
gsi_metguess_mod*init_: 3D-MET STATE VARIABLES:
u
v
div
vor
tv
q
oz
cw
......
```

```
&SETUP
GENCODE = 78.0000000000000 ,
FACTQMIN = 0.000000000000000E+000 ,
FACTQMAX = 0.000000000000000E+000 ,
......
```

2. Read in the background field

The following lines in stdout immediately following the namelist section, indicate that GSI is reading the background fields. For regional GSI observer run, background file wrf_inout is used and therefore users will see some information about problems with files wrf_inou1, ..., wrf_inou9, which is normal.

```
NetCDF error: No such file or directory
NetCDF error: No such file or directory
NetCDF error in wrf_io.F90, line 1078
NetCDF error in wrf_io.F90, line 1078
CONVERT_NETCDF_MASS: problem with flml = wrf_inou1, Status = -1021
iy,m,d,h,m,s= 2014 2 13 0 0 0
......
```

```
READ_wrf_massFILES: analysis date,minutes 2014 2 13 0 0 18996480
```

46
5. Applications for Regional and Global EnKF

READ_wrf_mass_FILES: sigma guess file, nming2 0.0000000000000000 2014 2 13 0 0 18996480
READ_wrf_mass_FILES: sigma fcst files used in analysis : 2 1.0000000000000000
READ_wrf_mass_FILES: surface fcst files used in analysis: 2 1.0000000000000000

3. Read in observational data

The stdout (ensemble mean) and stdout_memnnn (ensemble members) contains distinct differences for the reading in the observational data controlled by the lread_obs_save and lread_obs_skip options in the namelist

a. Ensemble mean

```
readObs_check: bufr file date is 2014021300 prepbufr q
readObs_check: bufr file date is 2014021300 prepbufr ps
readObs_check: bufr file uv not available satwndbufr
readObs_check: bufr file rv not available radarbufr
readObs_check: bufr file pcp_tmi trmm not available tmirrbufr
readObs_check: bufr file hirs3 n17 not available hirs3bufr

... ...
number of extra processors 1
READ_OBS: read 33 mhs mhs_n18 using ntasks= 2 0 1796 999999
READ_OBS: read 34 mhs mhs_n19 using ntasks= 2 2 585 999999
READ_OBS: read 35 mhs mhs_netop-a using ntasks= 2 0 175 999999
READ_OBS: read 36 mhs mhs_netop-b using ntasks= 2 2 3 999999

... ...
READ_BUFRTOVS : file=mhsbufr type=mhs sis=mhs_n18 nread= 70080
ithin= 2 rmesh= 60.000000 isfcalc= 0 nkeep= 7665 ntask= 2

... ...
READ_PREPBUFR : file=prepbufr type=ps sis=ps nread= 22486
ithin= 0 rmesh= 120.000000 isfcalc= 0 nkeep= 16005 ntask= 1

... ...
READ_OBS: write collective obs selection info to obs_input.common
```

For the ensemble mean, the same procedure for observation processing as the GSI analysis is followed (read_obs_check, READ_OBS, READ_BUFRTOVS, READMEPS, READ_PREPBUFR ...). Finally, it is indicated that the collective observations selection information is written out for use with the ensemble members.

b. Ensemble members

For the stdout_memnnn, the observation processing steps are skipped and observation information is read in from the ensemble mean:

```
observer_set: read collective obs selection info from obs_input.common
```

4. Indication that the GSI has successfully run:

```
[000]gsisub(): : complete.
```

```
ENDING DATE-TIME OCT 29, 2018 22:51:51.865 302 MON 2458421
PROGRAM GSI_ANL HAS ENDED.
```

After looking over each section of the standard output files, it can be concluded that the GSI observer ran without issues. Note that because the outer loop was set to 0 (mitter=0) for the GSI observer, no minimization occurred and therefore no analysis results were produced.
5. Applications for Regional and Global EnKF

5.2 Running EnKF for Regional Applications

5.2.1 Run Script

Once the GSI observer has been run successfully, the next step is to setup the EnKF run script, comenkf_run_regional.ksh. This script is located under comGSIv3.7_EnKFv1.3/ush. This run script uses the diag files generated by the GSI observer script as observation input, and generates the EnKF analysis. Similar to the GSI observer script, several user-specific modifications need to be made:

- Set up batch queueing system
  ```bash
  #PBS -A P????????
  #PBS -l walltime=00:20:00
  #PBS -N cg02-case24-enkf-allobs
  #PBS -l select=2:ncpus=36:mpiprocs=36
  #PBS -q premium
  #PBS -o out.case24-enkf-allobs
  #PBS -j oe
  ```

- Set up number of processors and the job queue system used:
  ```bash
  GSIPROC=36
  ARCH='LINUX_PBS'
  ```

  One difference from the GSI observer script is that the number of processors used should be greater than at least the number of ensemble members for running the EnKF. In this case we have 20 ensemble members and have requested 32 cores.

- Set up the analysis time, fixed files, EnKF executable
  ```bash
  ANAL_TIME=2014021300
  WORK_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/enkf_2014021300
  diag_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/gsidiag_2014021300
  BK_ROOT=/scratch/${user}/casedata/enkf_arw_2014021300/bk
  BK_FILE=${BK_ROOT}/wrfarw.ensmean
  GSI_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3
  FIX_ROOT=${GSI_ROOT}/fix
  ENKF_EXE=${GSI_ROOT}/build/bin/enkf_wrf.x
  CRTM_ROOT=/scratch/${user}/CRTM_v2.3.0/
  ENKF_NAMELIST=${GSI_ROOT}/ush/comenkf_namelist.sh
  ```

- Set background file and location of the diag files (from GSI observer)
  ```bash
  diag_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/gsidiag_2014021300
  BK_FILE=${BK_ROOT}/wrfarw.ensmean
  ```

  Two modifications to note for the EnKF are the diag_ROOT and BK_FILE. The diag_ROOT points to the working directory where the GSI observer was run. This directory contains diag* files which will be linked to the EnKF working directory. BK_FILE points to the background ensemble mean.

- Select ensemble parameters
  ```bash
  NMEM_ENKF=20
  BK_FILE_mem=${BK_ROOT}/wrfarw
  NLONS=129
  ```

48
5. Applications for Regional and Global EnKF

NLATS=70
NLEVS=50
IF_ARW=.true.
IF_NMM=.false.

This section sets information about the ensemble, including the number of members, background ensemble members, domain specifications, dynamical core, and list of observations to be used in the EnKF. Note that the ensemble members (BK_FILE_mem) do not include the 3-digit member number. This will be appended to the name specified for this field later in the script.

- Select observations

  list="conv amsua_n18 mhs_n18 hirs4_n19"

This list is contained with the ensemble parameter section. This list of observations/platforms to be assimilated can be different from the list that run through the GSI observer but be aware that only observations/platforms that have been run through the GSI observer (and therefore a diag* file exists) are valid.

5.2.2 Run EnKF and Check Run Status

Once the run script is set up properly for the case and the machine, EnKF can be run through the run script. The following command will submit the job:

  $ qsub comenkf_run_regional.ksh

While the job is running, move to the working directory and check the details. Given the following working directory setup:

  WORK_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/enkf_2014021300

Go to directory /scratch/${user}/comGSIv3.7_EnKFv1.3/run and check the run directory. A directory named enkf_2014021300 should have been created. This directory is the run directory for this EnKF case study. The directory will be populated with many links:

- diag_conv_ges.ensmean
- diag_conv_ges.mem001
- diag_conv_ges.mem002
- diag_conv_ges.mem003
- diag_conv_ges.mem004
- diag_amsua_n18_ges.ensmean
- diag_amsua_n18_ges.mem001
- diag_amsua_n18_ges.mem002
- diag_amsua_n18_ges.mem003
- diag_amsua_n18_ges.mem004

49
5. Applications for Regional and Global EnKF

The diag files are linked from the GSI observer working directory. These links are specified in the run script. Note that there are broken links to diag_hirs4_n19_ges.* because hirs4_n19 is not the one that run through the GSI observer.

The first guess files are also linked into the directory from the run script, pointing to the path of the ensemble mean and ensemble members designated in the setup section.

Similar to running the GSI observer, many other static files are linked or copied to this run directory, such as:

- **enkf.nml**: EnKF namelist
- **convinfo**: data usage control for conventional data
- **satbias_in**: satellite bias correction coefficient file
- **satbias_pc**: satellite bias correction file for passive channels
- **satinfo**: data usage and channel control for satellite radiance data
- **ozinfo**: data usage control from ozone data
- **stdout**: EnKF standard output file

The presence of the standard output file in the directory suggest the EnKF run script has successfully set up a run environment for the EnKF, properly linking the first guess and diag files from the GSI observer, and the EnKF executable is running. Once EnKF has finished running, analysis files should be generated for each member as well as the ensemble mean:
5. Applications for Regional and Global EnKF

5.2.3 Check for Successful EnKF Completion

The presence of the analysis files for the ensemble mean and each member as well as the standard output file indicates that the EnKF has run without crashing, but does not necessarily indicate a successful analysis. It is important to check the stdout file in the run directory to make sure the EnKF completed each step without any obvious problems. The following are several important areas of the standard output file to check:

1. Starting the EnKF run:

   running on 36 processors ...

   * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   PROGRAM ENKF_ANAL HAS BEGUN. COMPILED 2011319.55 ORG: NP25
   STARTING DATE-TIME OCT 29,2018 23:04:54.421 302 MON 2458421

2. Check namelist has been properly read in and configuration is correct:

   namelist parameters:
   ----------------------
   &NAM_ENKF
   DATEIN = 2014021300
   DATAPATH = ./
   IASSIM_ORDER = 0
   NVARS = 5
   COVINFLATEMAX = 100.000000
   COVINFLATEMIN = 1.00000000
   DETERMINISTIC = T
   ...

3. Check analysis time and number of ensemble members, as well as the background information, and information about the observations, such as CONINFO, OZINFO, RADINFO, and also the air mass bias correction coefficients:

   analysis time 2014021300
   20 members
   WARNING: nvars and massbal_adjust are no longer used!
   They are inferred from the anavinfo file instead.
   number of background forecast times to be used for H(x) = 1
   first-guess forecast hours for observation operator = 06
   number of background forecast times to be updated = 1
   first-guess forecast hours for analysis = 06
   using multiplicative inflation based on Pa/Pb
   enkf_main: number of threads 1
   line ignored in convinfo due to use flag ps 132 0 -1
   OZINFO_READ: jpch_oz = 272
   1 sbuv6_n14 lev = 1 use = -1 pob = 0.240 gross = 1.000
     error = 1.000 b_oz = 10.000 pg_oz = 0.000
   2 sbuv6_n14 lev = 2 use = -1 pob = 0.490 gross = 1.000
     error = 1.000 b_oz = 10.000 pg_oz = 0.000
   ...
   RADINFO_READ: jpch_rad = 3208
   1 amsua_n15 chan = 1 var = 3.000 varch_cld = 20.000 use = 1
     ermax = 4.500 b_rad = 10.000 pg_rad = 0.000 icld_det = -2 icloud = 1 iaeros = -1
   2 amsua_n15 chan = 2 var = 2.200 varch_cld = 18.000 use = 1
     ermax = 4.500 b_rad = 10.000 pg_rad = 0.000 icld_det = -2 icloud = 1 iaeros = -1
   ...

51
5. Applications for Regional and Global EnKF

RADINFO_READ: guess air mass bias correction coefficients below

<table>
<thead>
<tr>
<th></th>
<th>AMSUA_n15_1</th>
<th>AMSUA_n15_1</th>
<th>AMSUA_n15_1</th>
<th>AMSUA_n15_1</th>
<th>AMSUA_n15_1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.499241</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.461710</td>
<td>0.046909</td>
</tr>
<tr>
<td>2</td>
<td>1.679784</td>
<td>0.000000</td>
<td>0.000000</td>
<td>10.938589</td>
<td>-4.478499</td>
</tr>
</tbody>
</table>

4. Statistics of the ensemble prior

Following many lines describing the bias correction coefficients, the inventory of the observation number, observation types from the input diag* files, and the time spent reading in each observation type, statistics of the ensemble priors fit to all observations are provided for each region (NH, TR), depending on the analysis domain:

innovation statistics for prior:

<table>
<thead>
<tr>
<th>region, obtype, nobs, bias, innov std, sqrt(S+R), sqrt(S), sqrt(R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH all pe 14083 -0.591E-02 0.860E+00 0.813E+00 0.601E+00 0.547E+00</td>
</tr>
<tr>
<td>TR all pe 50 -0.629E-01 0.422E+00 0.641E+00 0.367E+00 0.526E+00</td>
</tr>
<tr>
<td>NH all t 7766 -0.228E+00 0.163E+01 0.128E+01 0.480E+00 0.119E+01</td>
</tr>
<tr>
<td>TR all t 138 -0.130E+00 0.119E+01 0.119E+01 0.234E+00 0.117E+01</td>
</tr>
<tr>
<td>NH all uv 21680 0.251E+01 0.124E+01 0.124E+01 0.238E+00 0.116E+01</td>
</tr>
<tr>
<td>TR all uv 302 -0.379E+00 0.268E+01 0.268E+01 0.464E+00 0.220E+01</td>
</tr>
<tr>
<td>NH all q 2553 -0.323E-01 0.157E+00 0.157E+00 0.868E-01 0.119E+00</td>
</tr>
<tr>
<td>TR all q 68 -0.321E-01 0.156E+00 0.156E+00 0.868E-01 0.119E+00</td>
</tr>
<tr>
<td>NH all uv 21680 0.136E-01 0.295E+01 0.276E+01 0.140E+01 0.220E+01</td>
</tr>
</tbody>
</table>

This table should be checked in order to determine if the inflation is appropriate. The goal is to make the total ensemble spreads of priors ($\sqrt{S+R}$) match the innovations (innov std) as much as possible. We can see in certain regions, the differences are notable. These differences are associated with higher observation error ($\sqrt{R}$). Users should consider tuning of inflation and localization, which is typically determined using cases with multiple assimilation cycles. Refer to section 4.2 of this User’s Guide for more information on tuning.

5. Check the minimum and maximum values of the fields at each vertical level as a quick sanity check

Surface pressure (spressmn) min/max range: 678.6733984375000 1032.2247314453125

READGRIDDATA_ARW: u 1 -20.5522194 20.7263222
READGRIDDATA_ARW: u 2 -20.8787842 21.2031631
READGRIDDATA_ARW: u 3 -21.6104663 22.3617668

52
5. Applications for Regional and Global EnKF

READGRIDDATA_ARW: u 4  -22.6256084  24.0846481
READGRIDDATA_ARW: u 5  -24.0716515  26.4782734
READGRIDDATA_ARW: u 6  -25.8923721  32.8184052

6. Domain and observation partition:

npts = 9030
min/max number of points per proc = 222 283
time to do model space decomp = 1.2303239927859977E-003
nobstot = 54989
min/max number of obs per proc = 1527 1528
time to do ob space decomp = 3.6349199945107102E-004
sending out observation prior ensemble perturbations from root...

nobstot*nanals = 1099780
... took 3.9184329943964258E-003 secs
time in load_balance = 0.13166810100665316 on proc 0

The analysis variables and the observations are distributed to different processors. We can see in this case that the min/max number of points per processor are 222 and 283, respectively. Similarly, we can see that the min/max number of observations per processor are 1527 and 1528, respectively, indicating that the observations are well dispersed among the processors.

5.2.4 Diagnose EnKF Analysis Results

At the bottom of the standard output file, there are several output statistics and tables that are helpful for users to diagnose the quality of the EnKF analysis.

1. Spread inflation of the analysis ensemble

   global ps prior std. dev min/max = 0.215956211 1.72586155
   NH mean ps prior standard deviation = 0.578957915
   NH mean ps posterior standard deviation (before inflation) = 0.287781417
   NH mean ps posterior standard deviation (after inflation) = 0.546174467
   NH mean ps inflation = 2.43589163
   TR mean ps prior standard deviation = 0.311811596
   TR mean ps posterior standard deviation (before inflation) = 0.280985802
   TR mean ps posterior standard deviation (after inflation) = 0.308485359
   TR mean ps inflation = 1.12325037
   time in inflate_ens = 5.9949199930997565E-003 on proc 0

   This section is important for checking whether the values of the inflation are reasonable by looking at a summary of the maximum and minimum values. Also, viewing the regional averaged statistics before and after inflation. For this case we can see both the NH and TR show larger standard deviation after the inflation, which is more consistent with the prior standard deviation.

2. Spread inflation of the analysis ensemble

   innovation statistics for posterior:
   conventional obs
   region, obtype, nobs, bias, innov std dev, sqrt(S+R), sqrt(S), sqrt(R):
   NH    all ps  14083  0.818E-02  0.756E+00  0.566E+00  0.143E+00  0.547E+00

53
5. Applications for Regional and Global EnKF

After the EnKF analysis, it is important to check the innovation statistics, much like the prior fit to all observation.

5.3 Running GSI Observer for Global Applications

5.3.1 Run Script

With both GSI and EnKF compiled (note that a separate global executable must be generated for EnKF: enkf_gfs.x) and the background ensemble files and observations acquired, the next step is to work with the gsi observer run script, comgsi_run_gfs.ksh. The location of this script is under comGSIv3.7_EnKFv1.3/ush. This run script is the same as the one used for a GSI global analysis run, with a few specific options selected in order to loop through all the ensemble members and generate the ensemble observation priors for each member and the ensemble mean. For the global examples, users must choose a case that corresponds to the spectral resolution of the background ensemble. For this example, we will be using T62. In addition to the GSI observer specific options, other user-specific modifications need to be made:

- Set up batch queueing system

To run GSI with multiple processors, a job queueing head must be added to the comgsi_run_gfs.ksh script. The set up of the job queue is dependent on the machine and the job control system. Refer to the GSI User’s Guide, section 3.2.2, for more examples of the setup section of this script. The following example is setup to run on a Linux cluster with PBS:

```
#PBS -A P????????
```
5. Applications for Regional and Global EnKF

#PBS -l walltime=00:20:00
#PBS -N cg02-case25-observer-gfs
#PBS -l select=1:ncpus=4:mpiprocs=4
#PBS -q premium
#PBS -o out.case25-observer-gfs
#PBS -j oe

• Set up number of processors and the job queue system used. For this example, LINUX_PBS and 36 processors are used:

  GSI_PROC=36
  ARCH='LINUX_PBS'

• Set up the case data, analysis time, fix files, GSI executable, and CRTM coefficients:

Set up analysis time and select global case:

  ANAL_TIME=2014092918
  GFSCASE=enkf_glb_t62

Set up working directory, which will hold the analysis results (all ensemble members will be run in this directory). This directory must have the proper write permissions as well as enough space to hold the output.

  WORK_ROOT=/scratch/${user}/comGSIv3.7-EnKFv1.3/run/gsidiag_${GFS_CASE}

Set path to background ensemble

  BK_ROOT=/scratch/${user}/casedata/enkf_glb_t62/bkg

Set path to the observation directory and the PrepBUFR file within the observation directory. All observations to be assimilated should reside in this directory.

  OBS_ROOT=/scratch/${user}/casedata/enkf_glb_t62/obs
  PREPBUFR=${OBS_ROOT}/gdas1.t18z.prepbufr.nr

Set the GSI system used for this case, including the paths of the fix files and the CRTM coefficients as well as the location of the GSI executable:

  CRTM_ROOT=/scratch/${user}/CRTM_v2.3.0
  GSI_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/
  FIX_ROOT=${GSI_ROOT}/fix
  GSI_EXE=${GSI_ROOT}/build/bin/gsi.x
  GSI_NAMELIST=${GSI_ROOT}/ush/comgsi_namelist_gfs.sh

Note that the GSI namelist generation script used for the global case (comgsi_namelist_gfs.sh) is different from the one used for the regional example.

• Choose to run GSI observer, set up background ensemble information

  if_observer=Yes  # Yes , or , No -- case sensitive!!!
  no_member=10
  PDYa='echo $ANAL_TIME | cut -c1-8'
  cyca='echo $ANAL_TIME | cut -c9-10'
  gdate='date -u -d "$PDYa $cyca -6 hour" +%Y%m%d%H'  # guess date is 6hr ago
  BK_FILE_mem=${BK_ROOT}/sfg_${gdate}

The option if_observer=Yes is the switch that enables comgsi_run_gfs.ksh to run the GSI observer (rather than GSI analysis). In this example, 10 ensemble members are selected with the naming convention: sfg_${GUESS_DATE}_memnnnn. Note that

55
### 5. Applications for Regional and Global EnKF

_memnnnnnn_ which is associated with each ensemble member (nnnn), is not included and will be appended later in the script.

- **Set the JCAP resolution for the case:**
  ```bash
  if [[ "${GFSCASE}" = "T62" ]]; then
    JCAP=62
    JCAP_B=62
  elif [[ "${GFSCASE}" = "T126" ]]; then
    JCAP=126
    JCAP_B=126
  elif [[ "${GFSCASE}" = "enkf_glb_t62" ]]; then
    JCAP=62
    JCAP_B=62
  elif [[ "${GFSCASE}" = "T254" ]]; then
    JCAP=254
    JCAP_B=574
  elif [[ "${GFSCASE}" = "T574" ]]; then
    JCAP=574
    JCAP_B=1534
  else
    echo "INVALID case = ${GFSCASE}"
    exit
  fi
  ``
  LEVS=64

  Note that this selection is filled based on selection of the GFSCASE. This example used the enkf_glb_t62 case. All cases use 64 levels.

  Further in the run script `comgsi_run_gfs.ksh`, the resolution parameters are set based on the requested case (resolution). For this case, we are using JCAP=62:
  ```bash
  elif [[ "${JCAP}" = "62" ]]; then
    LONA=192
    LATA=128
    DELTIM=1200
    resol=2
  ```

- **Link satellite bias correction coefficients, background ensemble files, and observations:**
  ```bash
  if [[ "${GFSCASE}" = "enkf_glb_t62" ]]; then
    cp ${BK_ROOT}/bfg_${gdate}_fhr03_ensmean ./sfcf03
    cp ${BK_ROOT}/bfg_${gdate}_fhr06_ensmean ./sfcf06
    cp ${BK_ROOT}/bfg_${gdate}_fhr09_ensmean ./sfcf09
    cp ${BK_ROOT}/sfg_${gdate}_fhr03_mem001 ./sigf03
    cp ${BK_ROOT}/sfg_${gdate}_fhr06_mem001 ./sigf06
    cp ${BK_ROOT}/sfg_${gdate}_fhr09_mem001 ./sigf09
  else
    ...
  fi
  ```
  ```bash
  cp ${GSI_ROOT}/fix/comgsi_satbias_in ./satbias_in
  cp ${GSI_ROOT}/fix/comgsi_satbias_pc_in ./satbias_pc_in
  ...
  # Link to the prepbufr data
  ln -s ${PREPBUFR} ./prepbufr
  ```
  ```bash
  # Link to the other observation data
  if [ -r "${OBS_ROOT}/satwnd" ]; then
  ```
Past the arch selection, environment variable checks, and creation of working directory, users will find the location where the observations are linked. For this case, we can see that the conventional PrepBUFR observations have been linked, as well as AMSU-A, HIRS4 and MHS satellite radiance BUFR files. These files will be linked to the working directory and separate observation innovation (diag) files will be generated for each observation.

5.3.2 Run GSI Observer and Check Run Status

Once the run script is set up properly for the case and the machine, GSI can be run through the run script. The following command will submit the job:

```
$ qsub comgsi_run_gfs.ksh
```

While the job is running, move to the working directory and check the details. Given the following working directory setup:

```
WORK_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/gsidiag_${GFSCASE}
```

Go to directory `/scratch/${user}/comGSIv3.7_EnKFv1.3/` and check the run directory. A directory named `gsidiag_enkf_g1b_t62` should have been created. This directory is the run directory for this GSI observer case study. The directory will be populated with many files such as:

```
AerosolCoeff.bin  hirs4_n19.TauCoeff.bin  sndrD1_g14.TauCoeff.bin
ahi_himawari8.SpcCoeff.bin  iasi616_metop-a.SpcCoeff.bin  sndrD1_g15.SpcCoeff.bin
ahi_himawari8.TauCoeff.bin  iasi616_metop-a.TauCoeff.bin  sndrD1_g15.TauCoeff.bin
airs281SUBSET_aqua.SpcCoeff.bin  iasi616_metop-b.SpcCoeff.bin  sndrD2_g11.SpcCoeff.bin
... ...
```

These files are CRTM coefficients that have been linked to this run directory through the GSI run script. Similar to running the GSI analysis, many other files are linked or copied to this run directory, such as:

Additionally, for the GSI observer many files are generated as a result of the `lread_obs_save` and `lread_obs_skip` options in the namelist for writing/reading collective observation selection information:
5. Applications for Regional and Global EnKF

```plaintext
gsiparm.anl: GSI namelist
prepbufr : PrepBUFR file for conventional observation
convinfo : data usage control for conventional data
satbias_in: satellite bias correction coefficient file
satinfo  : data usage and channel control for satellite radiance data
berror_stats: background error file
errttable : observation error file
```

- **obs_input.nnnn**: During ensemble mean (save=True, skip=False), save all observation preprocessing to this file. Each file (nnn) is for a different variable type.
- **pennnn.obs_setup**: When looping through ensemble members (save=False, skip=True), all observation preprocessing is skipped, obs_input.nnnn files are read and pennnn.obs_setup is written for each processor (nnn) for all observations.

As the GSI observer is running for the ensemble mean as well as looping through each member, files are for each ensemble member, as well as the ensemble mean:

The presence of the standard output files in the directory suggest the GSI observer run scripts have successfully set up a run environment for the GSI observer, properly looping through the ensemble members, and the GSI executable is running on each ensemble member. Once GSI has finished running, diag files should be generated for each observation type for each member as well as the ensemble mean:

```
diag_conv_ges.ensmean diag_conv_ges.mem004 diag_conv_ges.mem008
diag_conv_ges.mem001 diag_conv_ges.mem005 diag_conv_ges.mem009
diag_conv_ges.mem002 diag_conv_ges.mem006 diag_conv_ges.mem010
diag_conv_ges.mem003 diag_conv_ges.mem007
```

- **pennnn.conv_01**: Files generated after O-B, generated for conventional observations as well as one file per sensor (e.g. AMSU-A n18). These files are trimmed for resulting diag files and cleaned.
- **list_run_directory**: Directory listing after ensemble mean is run, before directory is cleaned for running the ensemble members.
- **list_run_directory_memnnn**: As above, for each ensemble member (nnn).
- **stdout**: Standard output file for ensemble mean.
- **stdout_memnnn**: Standard output file for each ensemble member (nnn).

58
5. Applications for Regional and Global EnKF

5.3.3 Check for Successful GSI Completion

The presence of the diag files and standard out files for the ensemble mean and each member indicates that the GSI observer has run without crashing, but does not necessary indicate a successful analysis. It is important to check the stdout files in the run directory to make sure the GSI observer completed each step without any obvious problems. The following are several important areas of the standard out file to check:

1. Read in the anavinfo and namelist

   gsi_metguess_mod*init_: 2D-MET STATE VARIABLES:
   ps
   z
   gsi_metguess_mod*init_: 3D-MET STATE VARIABLES:
   u
   v

   * ... *

   PROGRAM GSI_ANL HAS BEGUN. COMPILED 1999232.55 ORG: NP23
   STARTING DATE-TIME OCT 29, 2018 22:51:30.694 302 MON 2458421

   &SETUP
   GENCODE= 82.0000000000000000 ,
   FACTQMIN= 5.0000000000000000 ,
   FACTQMAX= 5.0000000000000000 ,

2. Read in the background field

   The following lines in the stdout immediately following the namelist section, indicate that GSI is reading the background fields.

   GESINFO: Read NCEP sigio format file, sigf06
   READ_FILES: analysis date,minutes 2014 9 29 18 0 19325880
   READ_FILES: atm guess file sigf03 3.0000000000000000 12 9 29 2014 19325700
   READ_FILES: atm guess file sigf06 6.0000000000000000 12 9 29 2014 19325880
   READ_FILES: atm guess file sigf09 9.0000000000000000 12 9 29 2014 19326060
   READ_FILES: sfc guess file sfcf03 3.0000000000000000 12 9 29 2014 19325700
   READ_FILES: sfc guess file sfcf06 6.0000000000000000 12 9 29 2014 19325880
   READ_FILES: sfc guess file sfcf09 9.0000000000000000 12 9 29 2014 19326060
   READ_FILES: atm fcst files used in analysis: 3 6 9 0.00000 3.00000 6.000000 2
   READ_FILES: sfc fcst files used in analysis: 3 6 9 0.00000 3.00000 6.00000 2
   GESINFO: Guess date is 12 9 29 2014 6.00000000000000000
   GESINFO: Analysis date is 2014 9 29 18 0 2014092918 3.00000000000000000

59
3. Read in observational data

The stdout (ensemble mean) and stdout_memnnn (ensemble members) contains distinct differences for the reading in the observational data controlled by the lread_obs_save and lread_obs_skip options in the namelist.

a. Ensemble mean

```plaintext
read_obs_check: bufr file date is 2014092918 prepbufr q
read_obs_check: bufr file date is 2014092918 prepbufr ps
read_obs_check: bufr file uv not available satwndbufr
... ...
number of extra processors 2
READ_OBS: read 28 mhs mhs_n18 using ntasks= 2 0 780 999999 999999
READ_OBS: read 29 mhs mhs_metop-a using ntasks= 2 2 514 999999 999999
... ...
READ_OBS: write collective obs selection info to obs_input.common
```

For the ensemble mean, the same procedure for observation processing as the GSI analysis is followed (read_obs_check, READ_OBS, READ_BUFRTOVS, READ_GPS, READ_PREPBUFR, ...). Finally, it is indicated that the collective observations selection information is written out for use with the ensemble members.

b. Ensemble members

For the stdout_memnnn, the observation processing sets are skipped and observation information is read in from the ensemble mean:

```plaintext
OBSERVER_SET: read collective obs selection info from obs_input.common
```

4. Indication that the GSI observer has successfully run:

```plaintext
glbsoi: complete
[000]gsisub(): : complete.

ENDING DATE-TIME JUL 15, 2016 15:48:52.880 197 FRI 2457585
PROGRAM GSI_ANL HAS ENDED.
```

After looking over each section of the standard output files, it can be concluded that the GSI global observer ran without issues. Note that because the outer loop was set to 0 (miter=0) for the GSI observer, no minimization occurred and therefore no analysis results were produced.

5.4 Running EnKF for Global Applications

5.4.1 Run Script

Once the GSI global observer has been run successfully, the next step is to setup the EnKF run script, comenkf_run_gfs.ksh. This script is located under comGSIv3.7_EnKFv1.3/ush. This run script uses the diag files generated by the GSI global
5. Applications for Regional and Global EnKF

observer script as observation input, and generates the global EnKF analysis. Similar to the GSI observer script, several user-specific modifications need to be made:

- Set up batch queueing system

```
#PBS -A P??????
#PBS -l walltime=00:20:00
#PBS -N cg02-case26-enkf-gfs
#PBS -l select=2:ncpus=36:mpiprocs=36
#PBS -q premium
#PBS -o out.case26-enkf-gfs
#PBS -j oe
```

One difference from the GSI observer script is that the number of processors used should be greater than at least the number of ensemble members for running the EnKF. In this case we have 10 ensemble members and have requested 36 cores due to the large global domain.

- Set up the analysis time, global case, fixed files, EnKF executable

```
ANAL_TIME=2014092918
GFSCASE=enkf_glb_t62
WORK_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/enkf_${GFSCASE}
BK_ROOT=/scratch/${user}/casedata/enkf_glb_t62/bkg
GSI_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3
FIX_ROOT=${GSI_ROOT}/fix
ENKF_EXE=${GSI_ROOT}/build/bin/enkf_gfs.x
CRTM_ROOT=/scratch/${user}/CRTM_v2.3.0
ENKF_NAMELIST=${GSI_ROOT}/ush/comenkf_namelist_gfs.sh
```

- Set location of the diag files (from GSI observer)

```
DIAG_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/gsidiag_${GFSCASE}
```

One modifications to note for the EnKF is the `DIAG_ROOT`. The `DIAG_ROOT` points to the working directory where the GSI observer runs. This directory contains diag* files which will be linked to the EnKF working directory.

- Select ensemble and case parameters

```
NMEM_ENKF=10
LEVS=64
NVARS=5
```

```
... ... 
elif [[ "GFSCASE" = "enkf_glb_t254" ]]; then
  JCAP=254
  JCAP_B=254
... ...
elif [[ "JCAP" = "254" ]]; then
  LONA=512
  LATA=256
  DELTIM=1200
  resol=2
```

This section sets information about the number of background ensemble members, domain specifications. This information mirrors that of the `run_gsi_global.ksh` script.

- Read in namelist

61
5. Applications for Regional and Global EnKF

#Build EnKF namelist on-the-fly
  . $ENKF_NAMELIST

Similar to the regional enkf (comenkf_run_regional.ksh), the global EnKF builds the namelist from an outside script for namelist generation (enkf_namelist_gfs.sh).

- Select observations
  
  list="conv amsua_metop-a amsua_n18 amsua_n15"

  The above line is contained within the ensemble looping section. For the observations to be assimilated, be aware that only observations/platforms that have been run through the GSI observer (and therefore a diag* file exists) are valid. For this example case, we will run EnKF for conventional observations and satellite radiance from AMSU-A.

5.4.2 Run EnKF and Check Run Status

Once the run script is set up properly for the case and the machine, EnKF can be run through the run script. The following command will submit the job:

$ qsub comenkf_run_gfs.ksh

While the job is running, move to the working directory and check the details. Given the following working directory setup:

WORK_ROOT=/scratch/${user}/comGSIv3.7_EnKFv1.3/run/enkf_${GFSCASE}

Go to directory /scratch/${user}/comGSIv3.7_EnKFv1.3/ and check the run directory. A directory named enkf_enkf_glb_t62 should have been created. This directory is the run directory for this EnKF case study. The directory will be populated with many links:

diag_conv_ges.ensmean diag_conv_ges.mem004 diag_conv_ges.mem008
diag_conv_ges.mem001 diag_conv_ges.mem005 diag_conv_ges.mem009
diag_conv_ges.mem002 diag_conv_ges.mem006 diag_conv_ges.mem010
diag_conv_ges.mem003 diag_conv_ges.mem007
diag_amsua_metop-a_ges.ensmean diag_amsua_metop-a_ges.mem004 diag_amsua_metop-a_ges.mem006 diag_amsua_metop-a_ges.mem001 diag_amsua_metop-a_ges.mem005 diag_amsua_metop-a_ges.mem002 diag_amsua_metop-a_ges.mem003 diag_amsua_metop-a_ges.mem004 diag_amsua_metop-a_ges.mem005 diag_amsua_metop-a_ges.mem006 diag_amsua_metop-a_ges.mem007 diag_amsua_metop-a_ges.mem008 diag_amsua_metop-a_ges.mem009 diag_amsua_metop-a_ges.mem010
diag_amsua_n18_ges.ensmean diag_amsua_n18_ges.mem004 diag_amsua_n18_ges.mem008 diag_amsua_n18_ges.mem001 diag_amsua_n18_ges.mem005 diag_amsua_n18_ges.mem009 diag_amsua_n18_ges.mem002 diag_amsua_n18_ges.mem006 diag_amsua_n18_ges.mem007 diag_amsua_n15_ges.ensmean diag_amsua_n15_ges.mem004 diag_amsua_n15_ges.mem008 diag_amsua_n15_ges.mem001 diag_amsua_n15_ges.mem005 diag_amsua_n15_ges.mem009 diag_amsua_n15_ges.mem002 diag_amsua_n15_ges.mem006 diag_amsua_n15_ges.mem007 diag_amsua_n15_ges.mem003 diag_amsua_n15_ges.mem004 diag_amsua_n15_ges.mem005 diag_amsua_n15_ges.mem006 diag_amsua_n15_ges.mem007
5. Applications for Regional and Global EnKF

The diag files are linked from the GSI global observer working directory. These links are specified in the run script. Note that ozone diag files (gome, oni, sbuv) will appear in the directory without links (zero length files) if ozone diags are not present from the GSI observer.

sfg_2014092918_fhr06_ensmean sfg_2014092918_fhr06_mem004 sfg_2014092918_fhr06_mem008
sfg_2014092918_fhr06_mem001 sfg_2014092918_fhr06_mem005 sfg_2014092918_fhr06_mem009
sfg_2014092918_fhr06_mem002 sfg_2014092918_fhr06_mem006 sfg_2014092918_fhr06_mem010
sfg_2014092918_fhr06_mem003 sfg_2014092918_fhr06_mem007

The first guess files are also linked into the directory from the run script, pointing to the path of the ensemble mean and ensemble members designated in the setup section.

Similar to running the GSI observer, many other static files are linked or copied to this run directory, such as:

- **enkf.nml**: EnKF namelist
- **convinfo**: data usage control for conventional data
- **satbias_in**: satellite bias correction coefficient file
- **satbias_angle**: satellite bias correction angle file
- **satinfo**: data usage and channel control for satellite radiance data
- **ozinfo**: data usage control from ozone data
- **stdout**: EnKF standard output file

The presence of the standard output file in the directory suggests the EnKF run script has successfully set up a run environment for the EnKF, properly linking the first guess and diag files from the GSI observer, and the EnKF executable is running. Once EnKF has finished running, analysis files should be generated for each member:

sanl_2014092918_mem001 sanl_2014092918_mem005 sanl_2014092918_mem009
sanl_2014092918_mem002 sanl_2014092918_mem006 sanl_2014092918_mem010
sanl_2014092918_mem003 sanl_2014092918_mem007
sanl_2014092918_mem004 sanl_2014092918_mem008

5.4.3 Check for Successful EnKF Completion

The presence of the EnKF analysis files for each member as well as the standard out file indicates that the EnKF has run without crashing, but does not necessary indicate a successful analysis. It is important to check the stdout file in the run directory to make sure the EnKF completed each step without any obvious problems. The following are several important areas of the standard out file to check:

1. Check namelist has been properly read in and configuration is correct:

   namelist parameters:
   ---------------------
   &NAM_ENKF
   DATEIN    = 2014092918,
5. Applications for Regional and Global EnKF

```plaintext
DATAPATH  = ./
IASSIM_ORDER= 0,
NVARS= 5,
COVINFLATEMAX= 100.00000 ,
COVINFLATEMIN= 1.00000000 ,
DETERMINISTIC=T,
SORTINC=T,
CORRLENGTHNH= 2000.00000 ,
CORRLENGTHTR= 2000.00000 ,
CORRLENGTHSH= 2000.00000 ,

2. Check analysis time, number of ensemble members, as well as the actual analysis variables and the background type.

   analysis time 2014092918
   10 members
   WARNING: nvars and massbal_adjust are no longer used!
   They are inferred from the anavinfo file instead.
   number of background forecast times to be used for H(x) = 1
   first-guess forecast hours for observation operator = 06
   number of background forecast times to be updated = 1
   first-guess forecast hours for analysis = 06
   using multiplicative inflation based on Pa/Pb

3. Statistics of the ensemble prior

After many lines describing the bias correction coefficients, the inventory of the observation number, observation types from the input diag* files, and the time spent reading in each observation type, statistics of the ensemble priors fit to all observations are provided:

   innovation statistics for prior:
   conventional obs
   region, obtype, nobs, bias, innov stddev, sqrt(S+R), sqrt(S), sqrt(R):
   NH   all ps 11905 0.184E+00 0.107E+01 0.148E+01 0.462E+00 0.141E+01
   TR   all ps 2078 0.310E+00 0.122E+01 0.131E+01 0.273E+00 0.128E+01
   SH   all ps 895 -0.235E+00 0.135E+01 0.131E+01 0.752E+00 0.119E+01
   NH   all t 1798 0.690E-01 0.143E+01 0.208E+01 0.451E+00 0.203E+01
   TR   all t 275  0.390E-01 0.148E+01 0.212E+01 0.257E+00 0.211E+01
   SH   all t  70  0.155E+00 0.130E+01 0.185E+01 0.503E+00 0.178E+01
   NH   all uv 9776 0.552E-01 0.268E+01 0.238E+01 0.866E+00 0.221E+01
   TR   all uv 2508 -0.396E-01 0.208E+01 0.212E+01 0.257E+00 0.221E+01
   SH   all uv 1120  0.845E-01 0.205E+01 0.358E+01 0.114E+01 0.340E+01
   NH   all q 641  0.285E-01 0.137E+00 0.206E+00 0.506E-01 0.200E+00
   TR   all q  88  0.442E-01 0.108E+00 0.202E+00 0.246E-01 0.200E+00
   SH   all q  37  0.108E-01 0.117E+00 0.204E+00 0.409E-01 0.200E+00
   satellite brightness temp
   instrument, channel #, nobs, bias, innov stddev, sqrt(S+R), sqrt(S), sqrt(R):
   amsua_n15 1  27 -0.122E+00 0.387E+01 0.348E+01 0.176E+01 0.300E+01
   amsua_n15 2  26 -0.341E+00 0.375E+01 0.257E+01 0.132E+01 0.220E+01
   amsua_n15 3  27 -0.227E+00 0.240E+01 0.220E+01 0.176E+00 0.169E+01
   amsua_n15 4  27 -0.127E+00 0.570E+00 0.627E+00 0.181E+00 0.600E+00
   amsua_n15 5  27 -0.190E+00 0.271E+00 0.317E+00 0.103E+00 0.300E+00
   amsua_n15 7  41 -0.877E-01 0.213E+00 0.278E+00 0.121E+00 0.250E+00
   amsua_n15 8  41 -0.768E-01 0.211E+00 0.317E+00 0.158E+00 0.275E+00
   amsua_n15 9  41  0.128E+00 0.275E+00 0.381E+00 0.172E+00 0.340E+00
   amsua_n15 10 41  0.152E+00 0.320E+00 0.431E+00 0.159E+00 0.400E+00
   amsua_n15 12 41  0.735E+00 0.881E+00 0.103E+00 0.234E+00 0.100E+01
   amsua_n15 13 41  0.151E+01 0.160E+01 0.155E+01 0.400E+00 0.150E+01
   amsua_n15 15 26 -0.129E+01 0.476E+01 0.420E+01 0.231E+01 0.350E+01
```

64
This table should be checked in order to determine if the inflation is appropriate. 

As mentioned in the regional EnKF example, the goal is to make the total ensemble spreads of priors ($\sqrt{(S + R)}$) match the innovations ($innov \ std$) as much as possible. This is a global run using both conventional observation and satellite radiance data, we can see that all regions (NH,TR,SH) are listed for the conventional observations but not for radiances. We can see certain obtypes have fairly close values. On the other hand, there are still regions/obtypes that have noticeable differences, mainly associated with larger observation errors ($\sqrt{(R)}$). Users should consider tuning of inflation and localization, which is typically determined using cases with multiple assimilation cycles. Refer to section 4.2 of this User’s Guide for more information on tuning.

4. Check the minimum and maximum values of the first guess fields at each vertical level as a quick sanity check:

    | Time | First guess surface pressure |
    |------|-----------------------------|
    | 1    | 504.57253229632835          |
    | 2    | 502.231506                 |
    | 3    | 499.185242                 |

5.4.4 Diagnose EnKF Analysis Results

At the bottom of the standard output file, there are several output statistics and tables that are helpful for users to diagnose the quality of the EnKF analysis.

5. Statistics of the ensemble analysis

    global ps prior std. dev min/max = 4.96712141E-02 2.97986627
    NH mean ps prior standard deviation = 0.594350934
    NH mean ps posterior standard deviation (before inflation)= 0.422546178
    NH mean ps posterior standard deviation (after inflation) = 0.566656113

    |       | Global ps prior std. dev min/max | NH mean ps prior standard deviation | NH mean ps posterior standard deviation (before inflation) | NH mean ps posterior standard deviation (after inflation) |
    |-------|----------------------------------|-------------------------------------|----------------------------------------------------------|----------------------------------------------------------|
    |       | 4.96712141E-02 2.97986627        | 0.594350934                        | 0.422546178                                              | 0.566656113                                              |
5. Applications for Regional and Global EnKF

NH mean ps inflation = 1.39436281
SH mean ps prior standard deviation = 1.02483916
SH mean ps posterior standard deviation (before inflation) = 0.905294895
SH mean ps inflation = 1.15615737
TR mean ps prior standard deviation = 0.285711318
TR mean ps posterior standard deviation (before inflation) = 0.271452606
TR mean ps posterior standard deviation (after inflation) = 0.283461034
TR mean ps inflation = 1.03808999
time in inflate_ens = 8.0578343942761421E-002 on proc 0

This section is important for checking whether the values of the inflation are reasonable by looking at a summary of the maximum and minimum values. Also, viewing the regional averaged statistics before and after inflation. For this case we can see for all regions, larger standard deviation after the inflation are reported, which is more consistent with the prior standard deviation.

6. Spread inflation of the analysis ensemble

innovation statistics for posterior:

conventional obs

<table>
<thead>
<tr>
<th>region, obtype, nobs, bias, innov stddev, sqrt(S+R), sqrt(S), sqrt(R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NH all ps 11905 0.422E-01 0.903E+00 0.143E+01 0.230E+00 0.141E+01</td>
</tr>
<tr>
<td>TR all ps 2078 0.235E+00 0.113E+01 0.228E+00 0.128E+01</td>
</tr>
<tr>
<td>SH all ps 895 -0.183E+00 0.102E+01 0.126E+01 0.426E+00 0.119E+01</td>
</tr>
<tr>
<td>NH all t 1798 0.569E-01 0.205E+01 0.287E+00 0.203E+01</td>
</tr>
<tr>
<td>TR all t 275 0.354E+00 0.212E+01 0.206E+00 0.211E+01</td>
</tr>
<tr>
<td>SH all t 70 0.149E+00 0.367E+00 0.181E+01 0.178E+01</td>
</tr>
<tr>
<td>NH all uv 9776 0.334E-01 0.226E+01 0.227E+00 0.221E+01</td>
</tr>
<tr>
<td>TR all uv 2508 -0.578E-02 0.203E+00 0.203E+00 0.299E+00</td>
</tr>
<tr>
<td>SH all uv 1120 0.145E-01 0.347E+00 0.690E+00 0.340E+01</td>
</tr>
<tr>
<td>NH all q 641 0.334E-01 0.203E+00 0.203E+00 0.299E+00</td>
</tr>
<tr>
<td>TR all q 88 0.385E-01 0.178E-01 0.201E+00 0.200E+00</td>
</tr>
<tr>
<td>SH all q 37 0.149E-01 0.340E+00 0.340E+00 0.200E+00</td>
</tr>
</tbody>
</table>

satellite brightness temp

<table>
<thead>
<tr>
<th>instrument, channel #, nobs, bias, innov stddev, sqrt(S+R), sqrt(S), sqrt(R)</th>
</tr>
</thead>
<tbody>
<tr>
<td>amsua_n15 1 27 0.767E-01 0.349E+01 0.334E+01 0.146E+01 0.300E+01</td>
</tr>
<tr>
<td>amsua_n15 2 26 -0.242E+00 0.349E+01 0.247E+01 0.113E+01 0.220E+01</td>
</tr>
<tr>
<td>amsua_n15 3 27 -0.145E+00 0.213E+01 0.215E+01 0.796E+00 0.200E+01</td>
</tr>
<tr>
<td>amsua_n15 4 27 -0.990E-01 0.507E+00 0.573E+00 0.200E+00 0.600E+00</td>
</tr>
<tr>
<td>amsua_n15 5 27 -0.177E+00 0.314E+00 0.255E+00 0.915E-01 0.300E+00</td>
</tr>
<tr>
<td>amsua_n15 6 41 -0.108E+01 0.177E+00 0.264E+00 0.852E-01 0.250E+00</td>
</tr>
<tr>
<td>amsua_n15 10 41 0.126E+00 0.262E+00 0.262E+00 0.122E+01 0.400E+00</td>
</tr>
<tr>
<td>amsua_n15 12 41 0.126E+00 0.203E+00 0.203E+00 0.100E+01 0.200E+00</td>
</tr>
<tr>
<td>amsua_n15 13 41 0.126E+00 0.203E+00 0.203E+00 0.100E+01 0.400E+00</td>
</tr>
<tr>
<td>amsua_n15 24 0.126E+00 0.203E+00 0.241E+00 0.122E+01 0.100E+01</td>
</tr>
<tr>
<td>amsua_n15 5 27 -0.177E+00 0.314E+00 0.255E+00 0.915E-01 0.300E+00</td>
</tr>
<tr>
<td>amsua_n15 7 41 -0.683E-01 0.177E+00 0.264E+00 0.852E-01 0.250E+00</td>
</tr>
<tr>
<td>amsua_n15 27 0.126E+00 0.262E+00 0.262E+00 0.122E+01 0.400E+00</td>
</tr>
<tr>
<td>amsua_n18 1 24 0.145E-01 0.373E+01 0.373E+01 0.139E+01 0.250E+00</td>
</tr>
<tr>
<td>amsua_n18 10 41 0.126E+00 0.203E+00 0.203E+00 0.100E+01 0.200E+00</td>
</tr>
<tr>
<td>amsua_n18 1 24 0.145E-01 0.373E+01 0.373E+01 0.139E+01 0.250E+00</td>
</tr>
<tr>
<td>amsua_n18 12 41 0.145E-01 0.144E+01 0.144E+01 0.343E+00 0.150E+01</td>
</tr>
<tr>
<td>amsua_n18 13 41 0.145E-01 0.144E+01 0.144E+01 0.343E+00 0.150E+01</td>
</tr>
<tr>
<td>amsua_n18 15 26 -0.108E+01 0.431E+01 0.431E+01 0.030E+01 0.300E+00</td>
</tr>
<tr>
<td>amsua_n18 16 24 -0.923E+00 0.340E+01 0.340E+01 0.120E+01 0.220E+00</td>
</tr>
<tr>
<td>amsua_n18 17 24 -0.923E+00 0.340E+01 0.340E+01 0.120E+01 0.220E+00</td>
</tr>
<tr>
<td>amsua_n18 18 1 24 -0.923E+00 0.340E+01 0.340E+01 0.120E+01 0.220E+00</td>
</tr>
<tr>
<td>amsua_n18 3 24 -0.923E+00 0.340E+01 0.340E+01 0.120E+01 0.220E+00</td>
</tr>
<tr>
<td>amsua_n18 4 24 -0.923E+00 0.340E+01 0.340E+01 0.120E+01 0.220E+00</td>
</tr>
<tr>
<td>amsua_n18 5 24 -0.923E+00 0.340E+01 0.340E+01 0.120E+01 0.220E+00</td>
</tr>
<tr>
<td>amsua_n18 6 24 -0.923E+00 0.340E+01 0.340E+01 0.120E+01 0.220E+00</td>
</tr>
</tbody>
</table>
### 5. Applications for Regional and Global EnKF

<table>
<thead>
<tr>
<th>Variant</th>
<th>Cycle</th>
<th>Time (h)</th>
<th>$\sigma_{\text{G}1}$</th>
<th>$\sigma_{\text{G}2}$</th>
<th>$\sigma_{\text{G}3}$</th>
<th>$\sigma_{\text{G}4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>amsua_n18</td>
<td>15</td>
<td>24</td>
<td>-0.304E+01</td>
<td>0.553E+01</td>
<td>0.392E+01</td>
<td>0.176E+01</td>
</tr>
<tr>
<td>amsua_metop-a</td>
<td>1</td>
<td>21</td>
<td>-0.523E-01</td>
<td>0.320E+01</td>
<td>0.277E+01</td>
<td>0.119E+01</td>
</tr>
<tr>
<td>amsua_metop-a</td>
<td>2</td>
<td>22</td>
<td>0.169E+00</td>
<td>0.392E+01</td>
<td>0.243E+01</td>
<td>0.104E+01</td>
</tr>
<tr>
<td>amsua_metop-a</td>
<td>3</td>
<td>23</td>
<td>-0.324E+00</td>
<td>0.284E+01</td>
<td>0.215E+01</td>
<td>0.789E+00</td>
</tr>
<tr>
<td>amsua_metop-a</td>
<td>4</td>
<td>23</td>
<td>0.379E-03</td>
<td>0.805E+00</td>
<td>0.579E+00</td>
<td>0.180E+00</td>
</tr>
<tr>
<td>amsua_metop-a</td>
<td>5</td>
<td>23</td>
<td>-0.969E-01</td>
<td>0.227E+00</td>
<td>0.317E+00</td>
<td>0.103E+00</td>
</tr>
</tbody>
</table>

After the EnKF analysis, it is important to check the innovation statistics, as previously discussed in the regional test case.
EnKF Basic Concepts and Code

6.1 Basic Concepts (in the Form of EnSRF)

6.1.1 Analysis Variables

In theory, EnKF can use any of the model prognostic variables as analysis variables as long as there exist meaningful/clear correlations between the variables and observations. Typically, for hydrostatic global models, horizontal wind components, temperature, water vapor, surface pressure, and ozone are used as analysis variables. For non-hydrostatic mesoscale models, like WRF, the vertical component of wind, rain/cloud related water content variables, and surface variables could be used as additional analysis variables.
6.1.2 Update of Analysis Variables

The minimum error-variance estimate of the analyzed variables $X^a$ is given by the traditional Kalman filter update equation,

\[
X^a = X^b + K(y^o - HX^b)
\]

\[
K = P^bH^T(HP^bH^T + R)^{-1}
\]

Where

- $X^b$ an m-dimensional background model forecast (i.e., prior)
- $X^a$ an m-dimensional analyses at model grids (i.e., posterior)
- $y^o$ a p-dimensional set of observations
- $H$ the operators that convert the model state to the observation space
- $P^b$ the $mm$-dimensional background-error covariance matrix
- $R$ the $pp$-dimensional observation-error covariance matrix
- $K$ the Kalman gain

Here, the Kalman gain is a function of the multivariate covariances of the model state variables and observations and the operator matrix that relates the model state to the observations. This update process is basically similar to a simple optimal interpolation (OI) scheme, where the Kalman gain is set to static.

The update equations (6.1) and (6.2) can be solved using ensemble technique. The Kalman gain can be estimated and propagated using a set of ensemble forecasts. Expressing the model state vector of the analysis variables as an ensemble mean (denoted by an overbar) and a deviation from the mean (denoted by a prime), the update equations for EnSRF ([6]) are written as:

\[
\bar{X}^a = \bar{X}^b + \bar{K}(y^o - H\bar{X}^b)
\]

\[
X'^a = X'^b - \bar{K}HX'^b
\]

\[
\bar{K} = \alpha K
\]

\[
\alpha = \left[ 1 + \sqrt{R/(HP^bH^T + R)} \right]^{-1}
\]

Where $K$ is the Kalman gain defined by (6.2) and estimated using the ensemble method described in the following section. $\bar{K}$ is the gain used to update ensemble deviations from the ensemble mean. Here, the observational error covariance is assumed uncorrelated, that is, $R$ is diagonal. Then observations can be assimilated serially, one at a time, so that the analysis after assimilation of the Nth observation becomes the background estimate for assimilating the (N+1)th observation. For an individual observation, $R$ and $HP^bH^T$ are scalars, $K$ and $\bar{K}$ are vectors of the same dimension as the model state vector (before applying localization). Both $K$ and $\bar{K}$ are calculated from the prior ensemble of the observation being assimilated and each of the analyses variables on the model grids individually.

Please note EnSRF is a revised EnKF that eliminates the necessity to perturb the observations. Therefore, the equation (6.4) does not contain the equivalent observation term to the one in the equation (6.3). However, this EnKF system itself provides an option to perturb the observations as well. In the future, we will add the description of the EnKF algorithm using perturbed observations into this user’s guide.
In the EnKF framework, there is no need to compute and store the full matrix \( P^b \). Instead, \( P^b H^T \) and \( H P^b H^T \) are estimated statistically from an ensemble of model forecasts/background. Specifically, these two quantities are obtained as:

\[
P^b H^T = \frac{1}{(n-1)} \sum_{i=1}^{n} X^b_i (H X^b_i)^T
\]

(6.7)

\[
H P^b H^T = \frac{1}{(n-1)} \sum_{i=1}^{n} H X^b_i (H X^b_i)^T
\]

(6.8)

where \( n \) is the ensemble size of model forecasts; \( i \) is the index of each individual ensemble member. The expected analyses error covariance at the model grids after assimilation is given by

\[
P^a = (I - KH) P^b (I - KH)^T
\]

(6.9)

6.1.3 Updates of Observation Priors

In a serial assimilation of observations, the model first-guess or backgrounds at model grids are updated by one single observation at a time. For the assimilation of next observation, the first-guess of the next observation (observation priors) needs to be re-computed using the updated model background and a forward observation operator. This process is straightforward while running on a single processor computer, but is not efficient in a parallel computing environment, particularly when the first-guess of the observations are all pre-calculated.

Alternatively, the first-guess of the next observation can also be updated by the observation being assimilated, similar to the update to the model variables, so that re-computing the first-guess using the observational operators is not needed (see [1]). After the update of the first guess of model variables, the \( N \)th observation being assimilated is also used to update the first-guess of the \((N+1)\)th and next observations within the localization distance. This process can be expressed in the following update equations, similar to the equations (6.3)-(6.6):

\[
\bar{Z}^a = \bar{Z}^b + K_z(y^o - H \bar{X}^b)
\]

(6.10)

\[
Z^a = Z^b - \alpha K_z H X^b
\]

(6.11)

\[
K_z = Z^b (H X^b)^T (HP^b H^T + R)^{-1}
\]

(6.12)

Where

\( y^o \) is the observation being assimilated.
\( Z^b \) is the first-guess of the next unassimilated observation
\( Z^a \) is the updated first-guess of the next unassimilated observation
\( K_z \) is the Kalman gain between the first-guess of the observation being assimilated and next unassimilated observation
6.1.4 Assimilation Order and Adaptive Thinning of Observations

In realistic assimilation systems, observations may have a non-linear relationship with the analysis variables. In addition, sampling errors are common due to the limited ensemble sizes. As a result, the achieved analysis can depend on the order of the observations assimilated. There are three options for choosing the orders of the observations for assimilation:

1. Assimilate observations in the order they are read in (default). This seems a reasonable choice for assimilating “BEST” observation types first (like radiosonde winds)
2. Shuffle the observations randomly before assimilating
3. Assimilate in order of increasing predicted observation analysis variance relative to the prior

For the third option, the predicted observational analysis variance against the first-guess in the observational space is defined as (for details, see [8] and [7]):

\[ \frac{HP^bH^T}{HP^bH^T} = R/(HP^bH^T + R) \] (6.13)

Note that the predicted variance is based on the observation priors’ variance as if the observation is assimilated alone, and does not include the effect of the assimilation of other observations.

The observations can be further thinned adaptively. It is done via the updated estimation of the predicted analysis variance of next observation. If the predicted analysis variance for one observation is very close to the prior of this observation, the impact of this observation is expected to be very small and, therefore, it can be skipped. The threshold is set by the namelist parameter `paoverph_thresh`.

6.1.5 Ensemble Spread Inflation

EnSRF uses a multiplicative inflation to inflate analyses/posterior ensemble spread back to the one of first-guess. The amount of inflation is given at each analysis grid point by:

\[
\sigma^b = \sqrt{\frac{\sum_{i=1}^{n}(X'^b_i)^2}{n-1}} \\
\sigma^a = \sqrt{\frac{\sum_{i=1}^{n}(X'^a_i)^2}{n-1}} \\
r = \left( \frac{\beta \sigma^b - \sigma^a}{\sigma^a} + 1 \right) \\
rX'^a_i \rightarrow X'^{\text{infl}}_i
\] (6.14)

where \( \sigma^b \) is the prior/first-guess ensemble standard deviation; \( \sigma^a \) is posterior/analyses ensemble standard deviation (before inflation); \( r \) is the inflation factor applied to each ensemble member deviation from the ensemble mean; \( \beta \) is a tunable namelist parameter (analpertwt)
6. EnKF Basic Concepts and Code Structure

defined in module params: If $\beta = 1$, ensemble is inflated so posterior standard deviation becomes the same as prior; If $\beta = 0$, there is no inflation.

For a given value of $\beta$, the inflation factor is proportional to the amount of ensemble spread reduction by assimilation of observations, normalized by the analyses ensemble spread. As a result, the inflation is in general larger where observations are denser or have larger impact. The actual inflation factor can be quite different for each variable and cross vertical and horizontal grids. If the smoothing namelist parameter ($\text{smoothparm}$) $> 0$, the estimated inflation factor is smoothed using a Gaussian spectral filter with an e-folding scale of $\text{smoothparm}$. The minimum and maximum values allowed can be controlled by namelist parameters. In additions, extra inflation can be obtained by adding random noise from a climatology distribution of the model errors ([7]). The amount of the random noises to be added can be controlled through namelist parameters as well.

The total amount of inflation from these two inflation schemes should meet the following relationship ([3]), as close as possible:

$$\langle (y^o - H\bar{X}^b)(y^o - H\bar{X}^b) \rangle = (HP^bH^T + R)$$

(6.15)

Satisfaction of this equation ensures that the total ensemble spreads (ensemble spreads plus observational error covariance, right side of the equation) is a reasonable estimation of the RMS errors of observation priors against observations (left side of the equation). This relationship justifies that the ensemble system works reasonably well. Moreover, the ensemble spreads should be tuned appropriately relative to observation errors (assuming these errors are correctly set). Substantially smaller ensemble spreads could lead to underweight of observations. As a result, EnKF may ignore the observations for assimilation and/or lead to an over-divergent ensemble.

6.1.6 Covariance Localization

To reduce the impact of spurious ensemble covariance on the update of both analyses variables at model grids and the first-guess of observations (observation priors), localization is applied to the covariance (Kalman gains) in the equations (6.7) and (6.12) in both horizontal and vertical. The function of [2] is used in horizontal localization. It uses a 5th order compact polynomial and the impact of observations is gradually reduced to zero at the specified cutoff distance. The scale height $-\log (P/P_{ref})$ is used in vertical localization.

The localization distance is an important tunable parameter for a successful analysis of EnSRF. Tuning the localization distance depends on several factors, including the ensemble size used, model grid resolutions, weather scenario, etc. In general, a larger ensemble size allow a longer localization distance. But assimilation with a smaller ensemble size may benefit from a reduced localization distance to reduce the impact of spurious covariance. In addition, assimilation of higher-resolution observations (e.g., radar data) may require a much shorter localization distance.
6.1.7 Adaptive Radiance Bias Correction with EnSRF

An adaptive radiance bias correction procedure is used for the satellite radiance data assimilation in EnSRF. The first-guess of the radiance observations are updated by the radiance observations using the assimilation procedure outlined in the section (6.2.2). The updated innovations (O-B) are then used to update the coefficients of the radiance bias correction scheme. The first-guess of the radiance observations are updated again using the updated bias correction coefficients. This process may be repeated/iterated multiple times until the updated first-guess of the radiance observations and bias correction coefficients converge.

6.2 EnSRF Code Structure and Key Functions

6.2.1 Main Code Tree

The code structure for main code *enkf_main.f90*:

<table>
<thead>
<tr>
<th>Step</th>
<th>Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>initialize</td>
<td>call mpi_initialize</td>
<td>initialize MPI</td>
</tr>
<tr>
<td></td>
<td>call init_rad</td>
<td>Initial radinfo variables</td>
</tr>
<tr>
<td></td>
<td>call read_namelist</td>
<td>read namelist (enkf.nml)</td>
</tr>
<tr>
<td></td>
<td>call mpi_initialize_io</td>
<td>initialize MPI communicator for IO tasks</td>
</tr>
<tr>
<td></td>
<td>call init_rad_vars</td>
<td>Initialize derived radinfo variables</td>
</tr>
<tr>
<td>prepare data</td>
<td>call getgridinfo</td>
<td>read horizontal grid information and pressure fields from forecast ensemble mean file</td>
</tr>
<tr>
<td></td>
<td>call readobs</td>
<td>Read observations, observation priors for each ensemble member (from diag** files generated by GSI forward operators). initial screening.</td>
</tr>
<tr>
<td></td>
<td>call print_innovstats</td>
<td>print innovation statistics for prior</td>
</tr>
<tr>
<td></td>
<td>if (readin_localization)</td>
<td>read in vertical profile of horizontal and vertical localization length scales, set values for each obs</td>
</tr>
<tr>
<td></td>
<td>call read_locinfo</td>
<td></td>
</tr>
<tr>
<td></td>
<td>call load_balance</td>
<td>do load balancing (partitioning of grid points, observations among processors)</td>
</tr>
<tr>
<td></td>
<td>call read_ensemble</td>
<td>read in prior ensemble members, distribute pieces to each task</td>
</tr>
<tr>
<td>analysis</td>
<td>if(letkf_flag) then</td>
<td>Update state variables, observation priors, and radiance bias correction coefficients with EnSRF or the EnKF with perturbed observations</td>
</tr>
<tr>
<td></td>
<td>call letkf_update</td>
<td></td>
</tr>
<tr>
<td></td>
<td>else</td>
<td></td>
</tr>
<tr>
<td></td>
<td>call enkf_update</td>
<td></td>
</tr>
<tr>
<td>write results</td>
<td>call inflate_ens</td>
<td>Inflate posterior ensemble using the multiplicative inflation scheme</td>
</tr>
<tr>
<td></td>
<td>call print_innovstats</td>
<td>print innovation statistics for posterior</td>
</tr>
<tr>
<td></td>
<td>call radinfo_write</td>
<td>write out bias coeffs on root</td>
</tr>
<tr>
<td></td>
<td>call write_ensemble</td>
<td>write out analysis ensemble</td>
</tr>
<tr>
<td>clean up</td>
<td>call obsmod_cleanup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>call gridinfo_cleanup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>call statevec_cleanup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>call loadbal_cleanup</td>
<td></td>
</tr>
<tr>
<td></td>
<td>call mpi_cleanup</td>
<td></td>
</tr>
</tbody>
</table>
6. EnKF Basic Concepts and Code Structure

Please note additive inflation may be added to the posterior ensemble offline.

6.2.2 Driver of Serial Ensemble Square Root Filter

If using the EnKF algorithm (if `letkf_flag = .false.`), the priors of model analyses variables, observation priors, and bias correction coefficients are updated by subroutine `enkf_update` in `enkf.f90`. The code structure of `enkf_update` is:

<table>
<thead>
<tr>
<th>Step</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: Determine the order of all observations for assimilation according to the namelist choice</td>
<td>if (iassim_order == 1) then ! create random index array so obs are assimilated in random order. else if (iassim_order .eq. 2) then ! assimilate obs in order of increasing HPaHT/HPbHT else ! assimilate obs in order they were read in end if</td>
</tr>
<tr>
<td>2 begin outer loop</td>
<td>do niter=1,numiter</td>
</tr>
<tr>
<td>2.1 reset ob error to account for gross errors</td>
<td>if (niter &gt; 1 .and. varqc) then !huber norm QC elif (huber) then !&quot;huber norm&quot; QC else !&quot;flat−tail&quot; QC. endif else ( \text{oberrvaruse(nob)} = \text{oberrvar(nob)} ) end if</td>
</tr>
<tr>
<td>2.2: assimilation loop over all observations, one observation at a time</td>
<td>obsloop: do nobx=1,nobstot .... end do obsloop ! loop over obs to assimilate</td>
</tr>
<tr>
<td>2.3</td>
<td>! make sure posterior perturbations still have ! zero mean ! distribute the O−A stats to all processors ! satellite bias correction update</td>
</tr>
<tr>
<td>2: end outer loop</td>
<td>enddo ! niter loop</td>
</tr>
</tbody>
</table>

The details of the step 2.2 is listed below:
6. EnKF Basic Concepts and Code Structure

<table>
<thead>
<tr>
<th>Step</th>
<th>Code in Step 2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>which observation to assimilate next</td>
</tr>
<tr>
<td></td>
<td>if (iassim_order == 2) then</td>
</tr>
<tr>
<td></td>
<td>else</td>
</tr>
<tr>
<td></td>
<td>nob = indxassim(nobx)</td>
</tr>
<tr>
<td></td>
<td>endif</td>
</tr>
<tr>
<td>2.</td>
<td>calculate :</td>
</tr>
<tr>
<td></td>
<td>[ \frac{H^\top H}{R}/(H^\top H + R) ]</td>
</tr>
<tr>
<td></td>
<td>hpfht = sum(anal_obchunk(:,nob1)**2)*r_nanalsml</td>
</tr>
<tr>
<td></td>
<td>hpfhtoberrinv = one/(hpfht+oberrvaruse(nob))</td>
</tr>
<tr>
<td></td>
<td>paoverpb = oberrvar(nob)/(hpfht + oberrvar(nob))</td>
</tr>
<tr>
<td>3.</td>
<td>calculate Equation 6.6</td>
</tr>
<tr>
<td></td>
<td>if (deterministic) then</td>
</tr>
<tr>
<td></td>
<td>! EnSRF.</td>
</tr>
<tr>
<td></td>
<td>obganl = -anal_obtmp/(one+sqrt(oberrvaruse(nob)*hpfhtoberrinv))</td>
</tr>
<tr>
<td></td>
<td>else</td>
</tr>
<tr>
<td></td>
<td>! perturbed obs EnKF.</td>
</tr>
<tr>
<td></td>
<td>end if</td>
</tr>
<tr>
<td>4.</td>
<td>Calculate Kalman gain ( kfgain ) ((K, Equation (6.5)) and add the analyses increments of the observation being assimilated to the ensemble mean ( \text{ensmean_chunk} ) and perturbations of analyses variables at model grids ( \text{anal_chunk} ) ((Equations (6.3) and (6.4)) )</td>
</tr>
<tr>
<td></td>
<td>if (nf2 &gt; 0) then</td>
</tr>
<tr>
<td></td>
<td>do ii=1,nf2 ! loop over nearby horiz grid points</td>
</tr>
<tr>
<td></td>
<td>do nb=1,nbackgrounds ! loop over background time levels</td>
</tr>
<tr>
<td></td>
<td>do nn=nn1,nn2</td>
</tr>
<tr>
<td></td>
<td>nnn=index_pres(nn)</td>
</tr>
<tr>
<td></td>
<td>if (taperv(nnn) &gt; zero) then</td>
</tr>
<tr>
<td></td>
<td>! gain includes covariance localization update all time levels</td>
</tr>
<tr>
<td></td>
<td>kfgain=taperv(nnn)*sum(anal_chunk(:,i,nn,nb)*anal_obtmp)</td>
</tr>
<tr>
<td></td>
<td>! update mean.</td>
</tr>
<tr>
<td></td>
<td>ensmean_chunk(i,nn,nb) = ensmean_chunk(i,nn,nb) + kfgain*obinc_tmp</td>
</tr>
<tr>
<td></td>
<td>! update perturbations.</td>
</tr>
<tr>
<td></td>
<td>anal_chunk(:,i,nn,nb) = anal_chunk(:,i,nn,nb) + kfgain*obganl(:,i)</td>
</tr>
<tr>
<td></td>
<td>end if</td>
</tr>
<tr>
<td></td>
<td>end do</td>
</tr>
<tr>
<td></td>
<td>end do ! end loop over background time levels.</td>
</tr>
<tr>
<td></td>
<td>end do ! end loop over nearby horiz grid points</td>
</tr>
<tr>
<td></td>
<td>end if ! if .not. lastiter or no close grid points</td>
</tr>
<tr>
<td>5.</td>
<td>Calculate the Kalman gain ( kfgain ) ((Kz, Equation (6.12)) and add the analyses increments of the observation being assimilated to the nearby observation priors ( \text{ensmean_obchunk, and anal_obchunk} ) ((Equations (6.10) and (6.11)).)</td>
</tr>
<tr>
<td></td>
<td>if (nf &gt; 0) then</td>
</tr>
<tr>
<td></td>
<td>do nob1=1,nf</td>
</tr>
<tr>
<td></td>
<td>! gain includes covariance localization.</td>
</tr>
<tr>
<td></td>
<td>kfgain = taper_disob(nob1)* &amp;</td>
</tr>
<tr>
<td></td>
<td>taper(lnsig<em>lnsig_linv)<em>taper(obt</em>obt_linv)</em> &amp;</td>
</tr>
<tr>
<td></td>
<td>sum(anal_obchunk(:,nob2)*anal_obtmp)*hpfht_con</td>
</tr>
<tr>
<td></td>
<td>! update mean.</td>
</tr>
<tr>
<td></td>
<td>ensmean_obchunk(nob2) = ensmean_obchunk(nob2) + kfgain*obinc_tmp</td>
</tr>
<tr>
<td></td>
<td>! update perturbations.</td>
</tr>
<tr>
<td></td>
<td>anal_obchunk(:,nob2) = anal_obchunk(:,nob2) + kfgain*obganl(:,i)</td>
</tr>
<tr>
<td></td>
<td>end do</td>
</tr>
</tbody>
</table>
The following are lists and explanations of the EnKF namelist variables. Users can also check file `params.f90` for the details.

## Section nam_enkf

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>datein</td>
<td>Analysis date in YYYYMMDDHH</td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td>datapath</td>
<td>path to data directory (include trailing slash)</td>
<td>Character(len=500)</td>
<td>””</td>
</tr>
<tr>
<td>iassim_order</td>
<td>= 0 for the order they are read in, =1 for random order, =2 for order of predicted posterior variance reduction (based on prior)</td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td>covinlatemax</td>
<td>maximum inflation</td>
<td>real(r_single)</td>
<td>1.0e30</td>
</tr>
<tr>
<td>covinlatemin</td>
<td>minimum inflation</td>
<td>real(r_single)</td>
<td>1.0</td>
</tr>
<tr>
<td>deterministic</td>
<td>if true, use EnSRF w/o perturbed obs. if false, use perturbed obs EnKF.</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td>sortinc</td>
<td>if false, re-order obs to minimize regression errors as described in Anderson (2003).</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td>corrlengthnh</td>
<td>length for horizontal localization (in km) in north hemisphere</td>
<td>real(r_single)</td>
<td>2800</td>
</tr>
<tr>
<td>corrlengthtr</td>
<td>length for horizontal localization (in km) in tropic</td>
<td>real(r_single)</td>
<td>2800</td>
</tr>
<tr>
<td>corrlengthsh</td>
<td>length for horizontal localization (in km) in south hemisphere</td>
<td>real(r_single)</td>
<td>2800</td>
</tr>
</tbody>
</table>
### A. Content of Namelist

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>varqc</td>
<td>Turn on varqc</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td>huber</td>
<td>use huber norm instead of &quot;flat-tail&quot;</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td>nlon</td>
<td>number of lons</td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td>nlat</td>
<td>Number of lats</td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td>smoothparm</td>
<td>smoothing parameter for inflation (-1 for no smoothing)</td>
<td>real(r_single)</td>
<td>-1</td>
</tr>
<tr>
<td>readin_localization</td>
<td>If true, read in localization length scales from an external file</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td>zhuberleft</td>
<td>Parameter for &quot;huber norm&quot; QC</td>
<td>real(r_single)</td>
<td>1.e30</td>
</tr>
<tr>
<td>zhuberright</td>
<td>Parameter for &quot;huber norm&quot; QC</td>
<td>real(r_single)</td>
<td>1.e30</td>
</tr>
<tr>
<td>optimelnh</td>
<td>observation time localization in hours over north hemisphere</td>
<td>real(r_single)</td>
<td>25.925</td>
</tr>
<tr>
<td>optimeltr</td>
<td>observation time localization in hours over tropic</td>
<td>real(r_single)</td>
<td>25.925</td>
</tr>
<tr>
<td>optimelsh</td>
<td>observation time localization in hours over south hemisphere</td>
<td>real(r_single)</td>
<td>25.925</td>
</tr>
<tr>
<td>reducedgrid</td>
<td>Do smooth in a reduced grid with a variable number on longitudes per latitude. The number of longitudes is chosen so that the zonal grid spacing is approximately the same as at the equator</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td>lnsigcutoffnh</td>
<td>length for vertical localization in ln(p) over north hemisphere for conventional observation</td>
<td>real(r_single)</td>
<td>2.0</td>
</tr>
<tr>
<td>lnsigcutofftr</td>
<td>length for vertical localization in ln(p) over tropic conventional observation</td>
<td>real(r_single)</td>
<td>2.0</td>
</tr>
<tr>
<td>lnsigcutoffsh</td>
<td>length for vertical localization in ln(p) over south hemisphere for conventional observation</td>
<td>real(r_single)</td>
<td>2.0</td>
</tr>
<tr>
<td>lnsigcutoffsatnh</td>
<td>length for vertical localization in ln(p) over north hemisphere for satellite radiance observation</td>
<td>real(r_single)</td>
<td>-999.0</td>
</tr>
<tr>
<td>lnsigcutoffsatr</td>
<td>length for vertical localization in ln(p) over tropic satellite radiance observation</td>
<td>real(r_single)</td>
<td>-999.0</td>
</tr>
<tr>
<td>lnsigcutoffsatsh</td>
<td>length for vertical localization in ln(p) over south hemisphere for satellite radiance observation</td>
<td>real(r_single)</td>
<td>-999.0</td>
</tr>
<tr>
<td>lnsigcutoffpsnh</td>
<td>length for vertical localization in ln(p) over north hemisphere for surface pressure observation</td>
<td>real(r_single)</td>
<td>-999.0</td>
</tr>
<tr>
<td>lnsigcutoffpsstr</td>
<td>length for vertical localization in ln(p) over tropic surface pressure observation</td>
<td>real(r_single)</td>
<td>-999.0</td>
</tr>
<tr>
<td>lnsigcutoffpssh</td>
<td>length for vertical localization in ln(p) over south hemisphere for surface pressure observation</td>
<td>real(r_single)</td>
<td>-999.0</td>
</tr>
</tbody>
</table>
### A. Content of Namelist

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>analpertwtnh</td>
<td>adaptive posterior inflation parameter over north hemisphere: 1 means inflate all the way back to prior spread</td>
<td>real(r_single)</td>
<td>0.0</td>
</tr>
<tr>
<td>analpertwtsh</td>
<td>adaptive posterior inflation parameter over tropic: 1 means inflate all the way back to prior spread</td>
<td>real(r_single)</td>
<td>0.0</td>
</tr>
<tr>
<td>analpertwttr</td>
<td>adaptive posterior inflation parameter over south hemisphere: 1 means inflate all the way back to prior spread</td>
<td>real(r_single)</td>
<td>0.0</td>
</tr>
<tr>
<td>sprd_tol</td>
<td>tolerance for background check: observations are not used if they are more than sqrt(S+R) from mean, where S is ensemble variance and R is observation error variance.</td>
<td>real(r_single)</td>
<td>9.9e31</td>
</tr>
<tr>
<td>nlevs</td>
<td>total number of levels</td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td>nanals</td>
<td>number of ensemble members</td>
<td>integer</td>
<td>0</td>
</tr>
<tr>
<td>nvars</td>
<td>number of 3d variables to update. For hydrostatic models, typically 5 (u,v,T,q,ozone).</td>
<td>integer</td>
<td>5</td>
</tr>
<tr>
<td>saterrfact</td>
<td>factor to multiply sat radiance errors</td>
<td>real(r_single)</td>
<td>1.0</td>
</tr>
<tr>
<td>univaroz</td>
<td>If true, ozone observations only affect ozone</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td>regional</td>
<td>If true, analysis is for regional ozone</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td>use_gfs_nemsio</td>
<td>If true, GFS background is in NEMS format</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td>paoverpb_thresh</td>
<td>if observation space posterior variance divided by prior variance less than this value, observation is skipped during serial processing. 1.0 = don’t skip any obs</td>
<td>(r_single)</td>
<td>1.0</td>
</tr>
<tr>
<td>latbound</td>
<td>definition of tropics and mid-latitudes (for inflation).</td>
<td>real(r_single)</td>
<td>25.0</td>
</tr>
<tr>
<td>delat</td>
<td>width of transition zone</td>
<td>real(r_single)</td>
<td>10.0</td>
</tr>
<tr>
<td>pseudo_rh</td>
<td>use ’pseudo-rh’ analysis variable, as in GSI</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td>numiter</td>
<td>number of times to iterate state/bias correction update. (only relevant when satellite radiances assimilated, i.e. nobs_sat&gt;0)</td>
<td>integer</td>
<td>1.0</td>
</tr>
</tbody>
</table>
## A. Content of Namelist

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>biasvar</strong></td>
<td>background error variance for rad bias coeffs (used in radbias.f90). Default is (old) GSI value. if negative, bias coeff error variance is set to - biasvar/N, where N is number of obs per instrument/channel. if newpc4pred is .true., biasvar is not used - the estimated analysis error variance from the previous cycle is used instead (same as in the GSI).</td>
<td>real(r_single)</td>
<td>0.1</td>
</tr>
<tr>
<td><strong>lupd_satbiasc</strong></td>
<td>if performing satellite bias correction update</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td><strong>cliptracers</strong></td>
<td>if true, tracers are clipped to zero when read in, and just before they are written out.</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td><strong>simple_partition</strong></td>
<td>partition obs for enkf using Graham’s rule</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td><strong>adp_anglebc</strong></td>
<td>turn off or on the variational radiance angle bias correction</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td><strong>angord</strong></td>
<td>order of polynomial for angle bias correction</td>
<td>Integer</td>
<td>0</td>
</tr>
<tr>
<td><strong>newpc4pred</strong></td>
<td>controls preconditioning due to sat-bias correction term</td>
<td>logical</td>
<td></td>
</tr>
<tr>
<td><strong>nmmb</strong></td>
<td>If true, ensemble forecast is NMMB</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td><strong>iau</strong></td>
<td>logical</td>
<td>false</td>
<td></td>
</tr>
<tr>
<td><strong>nhr_anal</strong></td>
<td>background forecast time for analysis</td>
<td>integer</td>
<td>6</td>
</tr>
<tr>
<td><strong>letkf_flag</strong></td>
<td>If true, do LETKF</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td><strong>boxsize</strong></td>
<td>Observation box size for LETKF (deg)</td>
<td>real(r_single)</td>
<td>90.0</td>
</tr>
<tr>
<td><strong>massbal_adjust</strong></td>
<td>mass balance adjustment for GFS</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td><strong>use_edges</strong></td>
<td>logical to use data on scan edges (.true.=to use)</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td><strong>emiss_bc</strong></td>
<td>If true, turn on emissivity bias correction</td>
<td>logical</td>
<td>false</td>
</tr>
</tbody>
</table>

### Section nam_wrf

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>arw</strong></td>
<td>regional dynamical core ARW</td>
<td>logical</td>
<td>false</td>
</tr>
<tr>
<td><strong>nmm</strong></td>
<td>regional dynamical core NMM</td>
<td>logical</td>
<td>true</td>
</tr>
<tr>
<td><strong>doubly_periodic</strong></td>
<td>logical</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>
### Section `satobs_enkf`

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>sattypes_rad</td>
<td>strings describing the satellite data type (which form part of the diag* filename).</td>
<td>character(len=20) array (nsat-max_rad)</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>dsis</td>
<td>strings corresponding to sattypes_rad which correspond to the names in the NCEP global_satinfo file.</td>
<td>character(len=20) array (nsat-max_rad)</td>
<td>&quot;&quot;</td>
</tr>
</tbody>
</table>

### Section `ozobs_enkf`

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Description</th>
<th>Data Type</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>sattypes_oz</td>
<td>strings describing the ozone satellite data type (which form part of the diag* filename)</td>
<td>character(len=20) array (nsat-max_oz)</td>
<td>&quot;&quot;</td>
</tr>
</tbody>
</table>
Bibliography


