JOINT CENTER FOR SATELLITE DATA ASSIMILATION

NO. 56, SUMMER 2017

JCSDA Quarterly

NOAA | NASA | US NAVY | US AIR FORCE

https://doi.org/10.7289/V54J0C9F

NEWS IN THIS QUARTER

SCIENCE UPDATE

All-sky Microwave Imager Data Assimilation at NASA GMAO

Satellite radiance observations combine global coverage with high temporal and spatial resolution and bring vital information to numerical weather prediction (NWP) analyses, especially in areas where conventional data are sparse. However, most satellite observations that are actively assimilated have been limited to clear-sky conditions due to difficulties associated with accounting for non-Gaussian error characteristics, nonlinearity, and the development of appropriate observation operators. To expand existing capabilities in satellite radiance assimilation, operational centers including the European Centre for Medium-range Weather Forecasts (ECMWF), United Kingdom's MetOffice, Japan Meteorological Agency (JMA), and National Centers for Environmental Prediction (NCEP) have been pursuing efforts to assimilate radiances affected by clouds and precipitation from microwave sensors. The expectation is that these data can provide critical constraints on meteorological parameters in dynamically sensitive regions and have positive impact on forecasts of precipitation.

As described in the previous issue of the *JCSDA Quarterly Newsletter*, NCEP's efforts to assimilate all-sky data in the Gridpoint Statistical Interpolation (GSI) system have been focused on temperature sounding data from the Advanced Microwave Sounding Unit-A (AMSU-A) and Advanced Technology Microwave Sounder (ATMS) in non-precipitating cloudy conditions. Efforts in all-sky satellite data assimilation at the Global Modeling and Assimilation Office (GMAO) at NASA Goddard Space Flight Center have been focused on the development of GSI configurations to assimilate all-sky data from microwave imagers such as the GPM Microwave Imager (GMI) and Global Change Observation Mission-Water (GCOM-W) Advanced Microwave Scanning Radiometer 2 (AMSR-2). Electromagnetic characteristics associated with their wavelengths allow microwave imager data to

(continued on page 2)

IN THIS ISSUE

	CLEAR-SKY	ALL-SKY
State variables	T, q, Ps, oz, Tskin, u, and v	T, q, Ps, oz, Tskin, u, v, ql, qi, qr, and qs
Analysis variables	Ψ, $X_{unblanced}$, $T_{unblanced}$, $Ps_{unbalanced}$, RH, oz, and Tskin	Ψ, $X_{unblanced}$, $T_{unblanced}$, $Ps_{unbalanced}$, RH, oz, Tskin, ql, qi, qr, and qs
Background Error	Ψ, $X_{unblanced}$, $T_{unblanced}$, $Ps_{unbalanced}$, RH, oz, and Tskin	Ψ, $X_{unblanced}$, $T_{unblanced}$, $Ps_{unbalanced}$, RH, oz, Tskin, ql, qi, qr, and qs (See Section 4.4)
Observation operator	CRTM (Version 2.2.3)	CRTM (Version 2.2.3) with a reconstructed cloud coefficient file of this study
Observation Error	Constant (and inflated during quality control process)	Symmetric error model (Geer and Bauer 2011)
Quality control	Screen out cloud affected radiance	Keep cloud and precipitation affected radiance Screen out data over ocean if SST < 278K
Bias correction predictors in VarBC	Constant, lapse rate, square of lapse rate, cosine of the zenith angle, and cloud liquid water path	Constant, lapse rate, lapse rate2, cos (zenith angle), Clavg, Clavg^2

Table 1: *Comparison of clear-sky and all-sky microwave TB assimilation framework in GEOS-5 ADAS. (T: atmospheric temperature, q: specific humidity, Tskin: skin temperature, Ps: surface pressure, oz: ozone mixing ratio, u: u-wind, v: v-wind, ql: liquid cloud mixing ratio, qi: ice cloud mixing ratio, qr: rain water mixing ratio, and qs: snow water mixing ratio, Ψ: stramfunction,* $X_{unblanced}$*: unbalanced velocity potential,* $T_{unblanced}$*: unbalanced temperature,* $Ps_{unbalanced}$*: unbalanced surface pressure, RH: relative humidity)*

be relatively transparent to atmospheric gases and thin ice clouds, and highly sensitive to precipitation. Therefore, GMAO's all-sky data assimilation efforts are primarily focused on utilizing these data in precipitating regions. The all-sky framework being tested at GMAO employs the GSI in a hybrid 4D-EnVar configuration of the Goddard Earth Observing System (GEOS) data assimilation system, which will be included in the next formal update of GEOS. This article provides an overview of the development of all-sky radiance assimilation in GEOS, including some performance metrics. In addition, various projects underway at GMAO designed to enhance the all-sky implementation will be introduced.

Highlights of all-sky satellite data configuration in GEOS

Various components of the GEOS system have been modified to assimilate cloud- and precipitation-affected microwave radiance data (Table 1). To utilize data in cloudy and precipitating regions, state and analysis variables have been added for ice cloud (qi), liquid cloud (ql), rain (qr) and snow (qs). This required enhancing the observation operator to simulate radiances in heavy precipitation, including frozen precipitation. Background error covariances in both the central analysis and EnKF analysis in hybrid 4D-EnVAR system have been expanded to include hydrometeors.

(continued on page 3)

JOINT CENTER FOR SATELLITE DATA ASSIMILATION

5830 University Research Court
College Park, Maryland 20740

Website: wwws.jcsda.noaa.gov

EDITORIAL BOARD

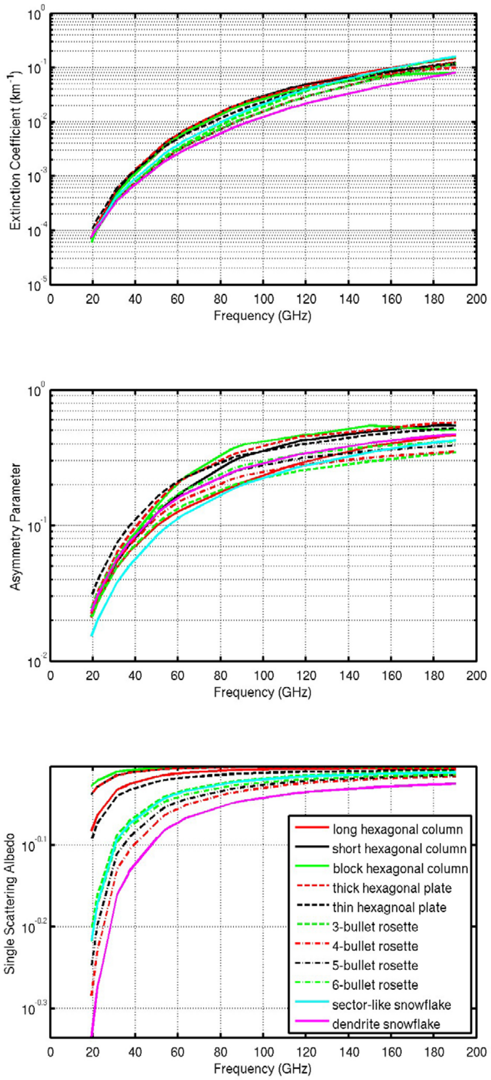
Editor:
James Yoe

Assistant Editor:
Biljana Orescanin

Director:
Thomas Auligné

Chief Administrative Officer:
James Yoe

Figure 1. *Microwave scattering properties as a function of frequency for a snow water content of 0.1 g m-3. In actual CRTM lookup table, extinction coefficients are stored in [m^{2}kg^{-1}] unit.*



In addition, the bias correction scheme was enhanced to reduce biases associated with thick clouds and precipitation.

a. Cloud scattering coefficients in CRTM

The observation operator for satellite radiances in GSI consists of spatial interpolation and the Community Radiative Transfer Model (CRTM), version 2.2.3. Scattering and extinction coefficients, asymmetry factor, and phase functions associated with hydrometeors for microwave wavelengths are read from a lookup table built using the Mie calculation for various cloud types (i.e., cloud ice, cloud liquid, rain, snow, graupel, and hail), and for various effective radius assuming a Gamma size distribution (Yang et al. 2005). Due to the known limitations of Mie scattering parameters for frozen hydrometeors, especially for high frequency (> 85 GHz) microwave channels (Kim 2006, Liu 2008, Geer and Baordo 2014), new parameters were calculated using the Discrete Dipole Approximation (DDA) method for non-spherical frozen precipitation from Liu (2008). Eleven different non-spherical ice crystal shapes in Liu's database, in addition to scattering properties of spherical ice crystals calculated with the Mie method, are examined to find an optimal choice of ice crystal shape to reconstruct the cloud-scattering coefficients for CRTM (Figure 1). For each shape of ice crystal, a CRTM cloud coefficient lookup table was generated for 33 microwave frequencies between 10.65 GHz and 190.31 GHz, seven atmospheric temperatures between 243 K and 303 K, and 405 effective radius sizes starting from 0.005 mm. The maximum effective radius considered for rain in the new CRTM coefficients is 1.191 mm. For snow crystals, the

(continued on page 4)

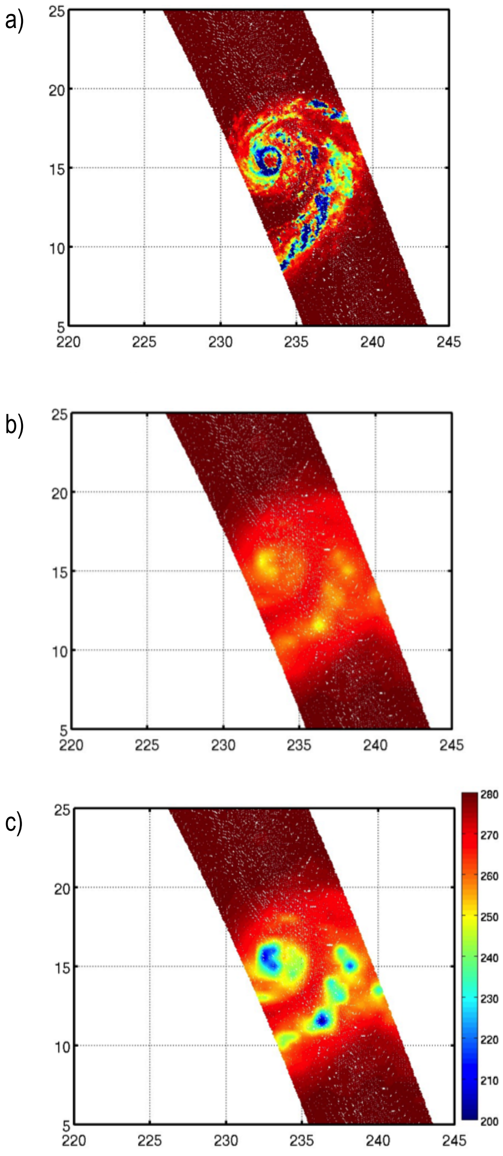


Figure 2. *Comparisons of simulated GMI 166 GHz vertically polarized brightness temperatures (TB) with the observations near Hurricane Celia on July 12, 2016 00Z: (a) Observed TBs, (b) CRTM simulated TBs with original scattering coefficients based on Mie method, and (c) CRTM simulated TBs with the DDA method calculated scattering properties of 3-bullet rosette snow crystals. The color bar shown in (c) works for (a) and (b) as well.*

maximum effective radius considered ranges from 0.664 mm to 1.278 mm, depending on snow crystal shape. Field et al. (2007) particle size distribution was assumed for frozen hydrometeors and Marshall-Palmer size distribution (Marshall and Palmer 1948) was assumed for liquid hydrometeors. After replacing original cloud coefficients with new cloud coefficients constructed with DDA scattering parameters, Simulated GMI brightness temperatures based on the new cloud coefficients are found to be closer to the observations and exhibit less first-guess departure bias in precipitating regions than those based on the original coefficients (Figure 2).

b. Enhanced bias correction

As with clear-sky radiances in the GEOS, bias correction for all-sky microwave radiances is performed using a variational bias correction scheme (VarBC, Dee 2004, Auligné et al. 2007) which estimates bias correction coefficients as part of the variational assimilation. For clear-sky microwave radiance data from microwave sensors such as AMSU-A, SSMIS, and ATMS, the bias predictors include a constant, the scan angle, a second-order polynomial of the atmospheric temperature lapse rate weighted by the radiance weighting function, and the retrieved cloud water path.

For the all-sky implementation, three changes were made to the original VarBC: First, the retrieved cloud liquid water path was removed as a predictor. Second, only near-clear sky observations with near-clear sky background are used in updating bias correction coefficients for pre-existing predictors. Third, the mean of the observed and calcu-

(continued on page 5)

lated cloud index based on 37-GHz brightness temperatures, (CI_{avg}), and its square, are used as two additional bias-correction predictors to correct the cloud amount-dependent first-guess biases. Results indicate that the modified VarBC scheme removes most of the bias in the first-guess departures, as indicated in Figure 3. The magnitude of remaining biases associated with thick cloud and heavy precipitation are reduced to less than 2 K in all CI ranges. Similar results are obtained for all GMI channels (not shown).

c. Background error covariance matrix

The analysis-control vector in the current GEOS analysis scheme includes stream function, unbalanced velocity potential, unbalanced virtual temperature, unbalanced surface pressure, relative humidity, ozone mixing ratio, and skin temperature (Rienecker et al. 2008). With the newly added control variables, the corresponding static and flow-dependent background error covariances must be generated. Climatological statistics were estimated following the NMC method (Parrish and Derber 1992) using pairs of 24-hour and 48-hour GEOS forecasts between June 1, 2016 and January 16, 2017. Ensemble covariances are based on the spread of the 32 ensemble forecasts from the GEOS hybrid scheme during each analysis cycle.

The panels on the left side of Figure 4 show the vertical distribution of the static background errors for cloud liquid, cloud ice, rain, and snow water. Aside from the fact that the estimated errors are by construction zonally invariant, they have generally

(continued on page 6)

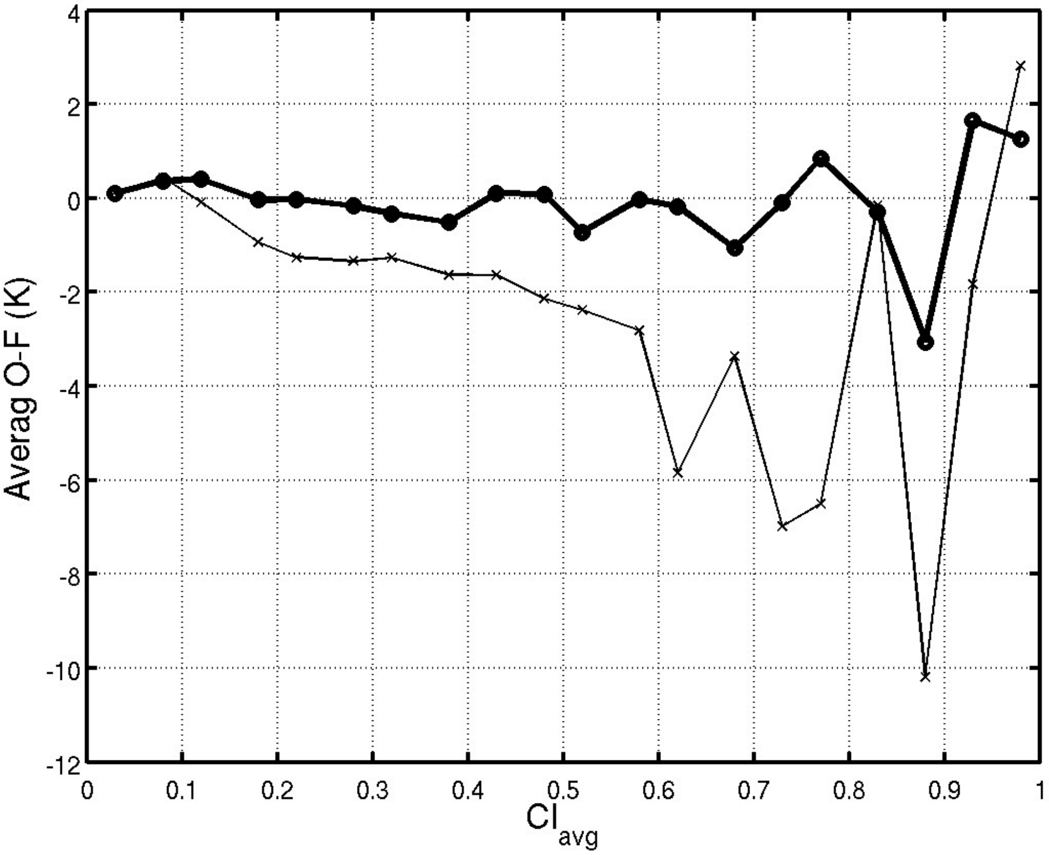
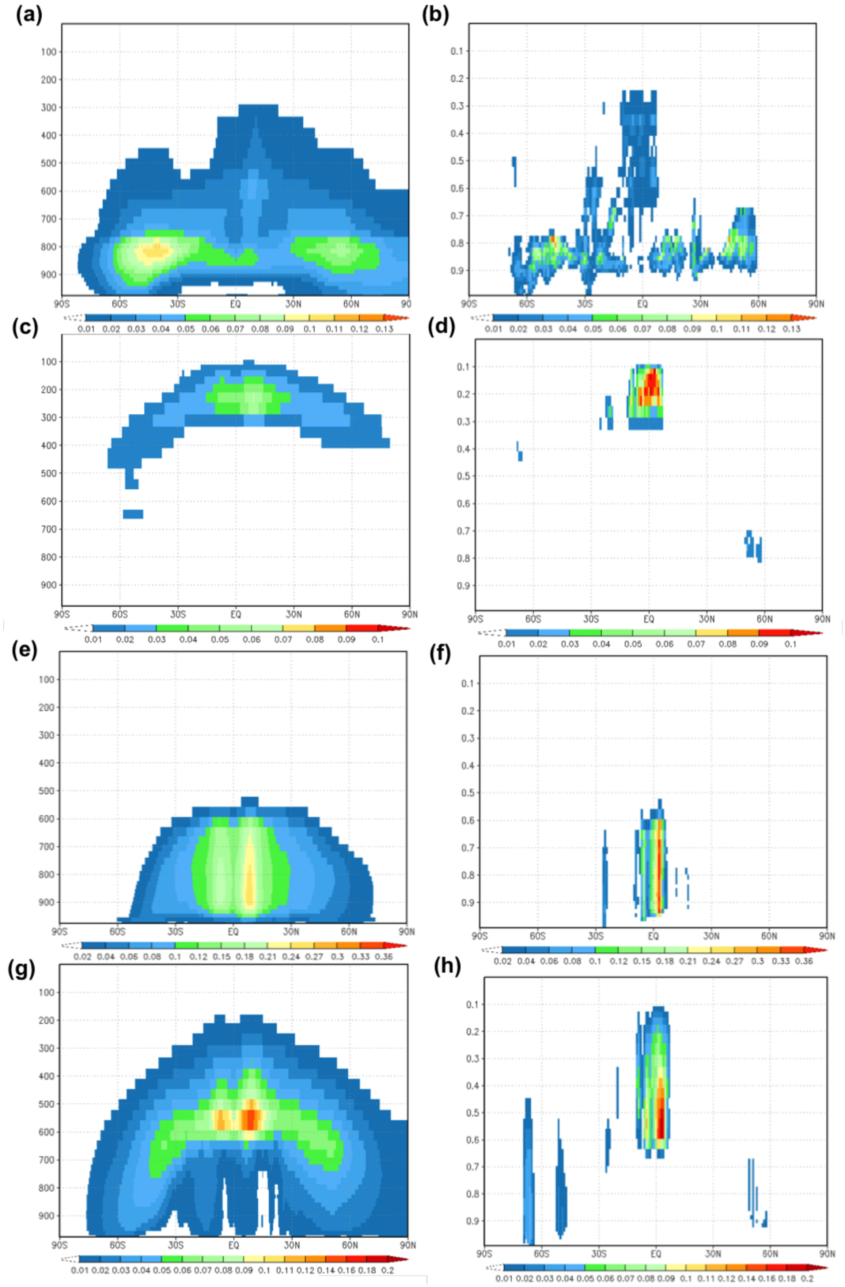


Figure 3. *Bias of first-guess departures of GMI channel 13 as a function of CIavg. Thin solid line shows the biases before, while thick solid line shows biases after, using* CI_{avg} *as additional predictors in VarBC. All assimilated data points between 12/01–12/31/2015 were used. Results only in the bins that have the number of data points greater than 5 are shown in this figure.*

***Figure 4:** Comparisons of static background errors (left figures) and ensemble background errors (right figures) as a function of latitude and vertical level. (a) and (b): liquid cloud, (c) and (d): ice cloud, (e) and (f): rain, and (g) and (h): snow.*



smooth spatial structure. Relatively large errors for liquid clouds are seen in storm tracks in midlatitudes. Static errors in the Southern Hemisphere are slightly larger than in the Northern Hemisphere. Generally speaking, the maximum errors for cloud liquid water occur in the layer between 900 hPa and 850 hPa. Static background errors for cloud ice water show large values near the tropical tropopause, where large amounts of cloud ice exist in the anvils of convective clouds. Static background errors for rain and snow are larger in the tropics than other latitudes. Large background errors for rain occur in the tropical lower troposphere between sea level and 600 hPa, while large errors for snow occur in the tropical middle troposphere between 600 hPa and 450 hPa.

(continued on page 7)

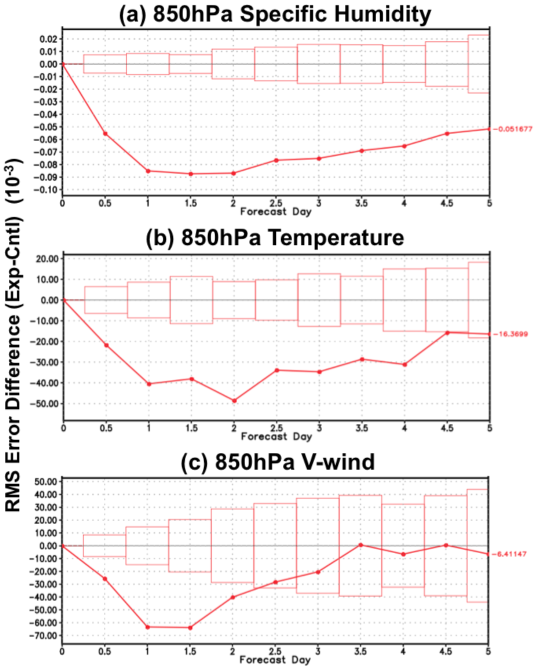


Figure 5: *RMS error differences between the GEOS control and all-sky GMI experiment in the tropics for December 2015: (a) 850hPa specific humidity (b) 850hPa temperature, (c) 850hPa V-wind.*

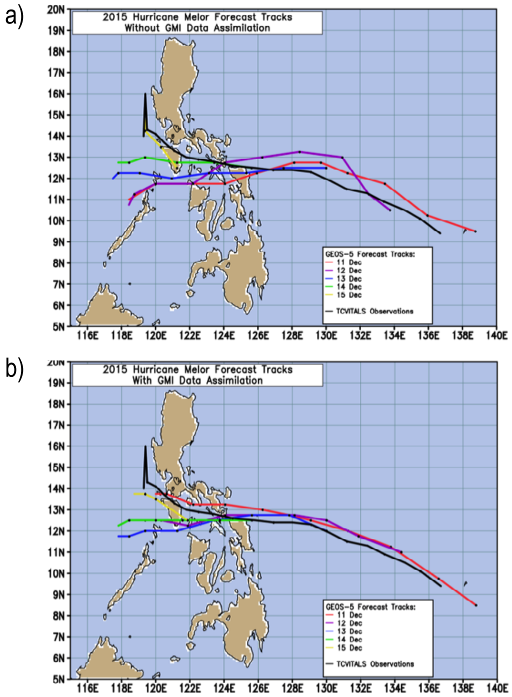


Figure 6: *GEOS forecasts of the track of hurricane Melor (December 2015): (a) without and (b) with the assimilation of all-sky GMI data.*

The panels on the right side of Figure 4 show cross-sections of ensemble background errors for hydrometeors taken from GEOS on 12 December 2015 12Z. The results indicate that the magnitudes of the ensemble background errors are similar to those of the static background errors, although ensemble-based estimates show more detailed flow-dependent structures. Note that there are regions with nearly zero ensemble error corresponding to areas where the ensemble members forecasted nearly zero clouds (clear sky). In contrast, the static background errors show nonzero values over broad ranges of latitude.

Impact Assessments

Cycled data assimilation experiments were conducted to examine the impact of all-sky GMI radiances on GEOS analyses and forecasts. GEOS was run in a hybrid 4D-EnVar configuration with a horizontal resolution of 0.5 degrees for the analysis and 0.25 degrees for the forecast. The control run assimilated all the data used routinely in GEOS (conventional data, AMSU-A, ATMS, MHS, IASI, AIRS, GPSRO, and satellite wind data), while the experimental run assimilated all-sky GMI data additionally. It was found that the all-sky GMI data generally have a significant impact on the lower tropospheric humidity and temperature analyses, especially in the tropics, which leads to improved forecasts of these quantities (Figure 5). Similar results were obtained for all seasons (not shown). In addition, a noticeable positive impact of all-sky GMI assimilation on hurricane track forecasts was identified for Hurricane Melor, which occurred in the western Pacific during December 2015 (Figure 6).

Discussion

Currently, static and ensemble background errors have the same weight (0.5) for all analysis variables in GEOS. Climatological background errors for highly nonlinear

(continued on page 8)

and situation-dependent clouds and precipitation may be less meaningful compared with other dynamical variables. To assign much larger weight to the ensemble-based background errors for hydrometeors, the capability to assign different weights for hydrometeors versus other dynamic variables is under development. The current all-sky framework will be enhanced by various updates both in the forecast model and analysis scheme. For example, the inclusion of a two-moment microphysics scheme (Barahona et al. 2014) in the GEOS forecast model will provide estimates of cloud particle size distributions to the all-sky observation operator. Future versions of the CRTM will account for cloud fraction in calculating radiances. This should improve the simulation of brightness temperature compared with the current version of CRTM, which considers only clear-sky or completely overcast conditions. In addition, we are testing various dynamic thinning approaches in order to use more data in cloudy and precipitating regions. All these enhancements are expected to extend the scope of all-sky radiance assimilation to include more microwave measurements and lead, in turn, to improved analyses and forecasts.

Min-Jeong Kim, Jianjun Jin, Will McCarty, Amal El Akkaroui, Ricardo Todling, Wei Gu, and Ron Gelaro (NASA Global Modeling and Assimilation Office)

References

Auligné, T., A.P. McNally, and D.P. Dee, 2007: Adaptive bias correction for satellite data in a numerical weather prediction system. *Q.J.R. Meteorol. Soc.*, 133, 631–642.

Barahona, D., A. Molod, J. Bacmeister, A. Nenes, A. Gettelman, H. Morrison, V. Phillips, and A. Eichmann, 2014: Development of two-moment cloud microphysics for liquid and ice within the NASA Goddard Earth Observing System Model (GEOS-5), *Geosci. Model Dev.*, 7, 1733–1766.

Dee, D.P., 2004: Variational bias correction of radiance data in the ECMWF system. *Proceedings of the ECMWF Workshop on Assimilation of High Spectral Resolution Sounders in NWP, Reading, UK, 28 June to 1 July 2004*. 97–112.

Field, P.R., A.J. Heymsfield, and A. Bansemer, A., 2007: Snow size distribution parameterization for midlatitude and tropical ice clouds, *J. Atmos*. Sci., 64, 4346–4365.

Geer, A.J., and F. Baordo, 2014: Improved scattering radiative transfer for frozen hydrometeors at microwave frequencies. *Atmos. Meas. Tech*. 7, 1839–1860.

Geer, A. J., and P. Bauer, 2011: Observation errors in all-sky data assimilation. *Q.J.R. Meterol. Soc.*, 137, 2024–2037. DOI:10.1002/qj.830

Kim, M.-J., 2006: Comparisons of single scattering approximations of randomly oriented ice crystals at microwave frequencies. *J. Geophys. Research*, 111 (D14201): [doi:10.1029/2005JD006892].;

Liu, G., 2008: A database of microwave single-scattering properties for nonspherical ice particles. *Bulletin of the American Meteor. Soc.*, Vol. 89, 1563–1570.

Marshall, J.S., and W.M.K. Palmer, 1948: The distribution of raindrops with size. *J. Meteor.*, 5, 165–166.

(continued on page 9)

Parrish, D.F., and J.C. Derber, 1992: The National Meteorological Center's spectral statistical interpolation analysis system. *Mon. Wea. Rev.*, 120, 1747–1763.

Rienecker, M.M., M. J. Suarez, R. Todling, J. Bacmeister, L. Takacs, H.-C. Liu, W. Gu, M. Sienkiewicz, R. D. Koster, R. Gelaro, I. Stajner, and J. E. Nielsen, 2008. The GEOS-5 Data Assimilation System—documentation of versions 5.0.1, 5.1.0, and 5.2.0. *Technical Report Series on Global Modeling and Data Assimilation*, Vol. 27, 1–118 pp.

Yang, P., H. Wei, H. Huang, B.A. Baum, Y.X. Hu, G.W. Kattawar, M.I. Mishchenko, and Q. Fu, 2005: Scattering and absorption property database for nonspherical ice particles in the near- through far-infrared spectral region. *Applied Optics*, 44, 5512–5523.

Introducing Object-Oriented Concepts into GSI

Enhancements are now being made to the Grid-point Statistical Interpolation (GSI) data assimilation system to expand its capabilities and open the way for broadening the scope of its applications. These represent a starting point for the so-called *GSI refactoring*, which is to take shape as part of the Joint Effort for Data-assimilation Integration (JEDI) project coordinated by Thomas Auligné and Yannick Trémolet of the JCSDA.

Our initial contribution amounts to introducing object-oriented concepts to: (1) improve the GSI handling and expandability of the various observation types associated with the forward observation operators (FOO), (2) generalize the interface to the background (guess) states interpolations needed by the FOO, and (3) provide a framework to allow for addition of user-specific subcomponents without need of changes to the actual GSI software. These implementations follow a bottom-up strategy so the refactored software never loses operability, thus remaining ready for use all along in currently supported GSI applications. A top-down approach to develop a Unified Forward Operator is currently underway, benefiting from contributions from many other collaborators; this, however, is not part of the present discussion.

Concepts of object-oriented (OO) programming have been around for a while and are supported by languages such as C++, Java, and many others. In general, OO programming enforces three main concepts: (1) abstraction; (2) encapsulation; and (3) polymorphism. From a scientific developer's point of view, *abstraction* allows for high-level code components to look simple, mimicking generic algorithmic concepts, similar to using mathematical symbols in equations; *encapsulation* allows the symbols to be defined in full details with separate expressions; and *polymorphism* allows for use of a single abstract entity with generic interfaces to represent specific entities of different types, while simultaneously managing encapsulated implementation differences between types as extensions. Fortran, the programming language of most codes used by our community, and in particular GSI, has slowly incorporated

(continued on page 10)

OO functionalities: Fortran 90 has supported abstraction and encapsulation since its initial stages. More recently, Fortran 2003 and 2008 bring in polymorphism, thus expanding the OO capabilities in the language.

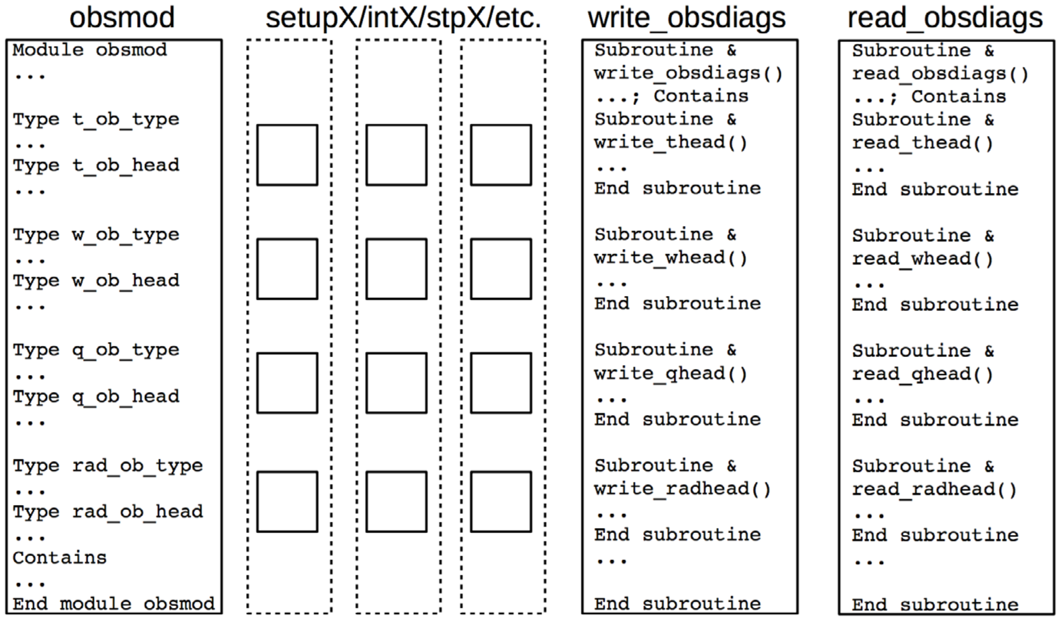
The path we chose to introduce advanced programming concepts in GSI represents a mild version of refactoring, where, to a large extent, the familiar code remains recognizable. What follows provides examples of how modifications are being made to GSI's observation handling, guess interpolators, and hooks to user-specific components.

Extensible Observation Types

At the time of this writing, the official release of GSI incorporates a polymorphic observation operator. At its initial stage, polymorphism has been introduced without strict encapsulation (see below). This has been done to allow users to familiarize themselves with the code changes before further changes take place, which will involve considerable shuffling of software. The original GSI implementation of the ability to handle multiple observation types can be labeled *procedural*, since the code is grouped by functionality instead of datatypes. This is illustrated in Figure 1, where *obsmod* controls initialization, finalization, and referencing to all observation types, whereas its "methods" (not treated as such in original code) are placed elsewhere and remain separated from other methods of the same type: the nonlinear observation operators (setupX), their linearized counterparts

(continued on page 11)

Figure 1. *Schematic view of original* ***procedural*** *coding of components related to the observation operators in GSI. Left-most rectangle (obsmod): initialization, finalization and general management of various observation types; middle rectangles: non-linear (setupX) and linearized observation operators (intX), and conjugate-gradient step calculations (stpX) for each observation type; two right-most rectangles: writes and reads of the observation types. Note the vertical (procedural) organization separates an observation type from its methods.*



```
Module obsmod
...

Type t_ob_type
...
Type t_ob_head
...

Type w_ob_type
...
Type w_ob_head
...

Type q_ob_type
...
Type q_ob_head
...

Type rad_ob_type
...
Type rad_ob_head
...
Contains
...
End module obsmod
```

```
Subroutine &
write_obsdiags()
...; Contains
Subroutine &
write_thead()
...
End subroutine

Subroutine &
write_whead()
...
End subroutine

Subroutine &
write_qhead()
...
End subroutine

Subroutine &
write_radhead()
...
End subroutine
...

End subroutine
```

```
Subroutine &
read_obsdiags()
...; Contains
Subroutine &
read_thead()
...
End subroutine

Subroutine &
read_whead()
...
End subroutine

Subroutine &
read_qhead()
...
End subroutine

Subroutine &
read_radhead()
...
End subroutine
...

End subroutine
```

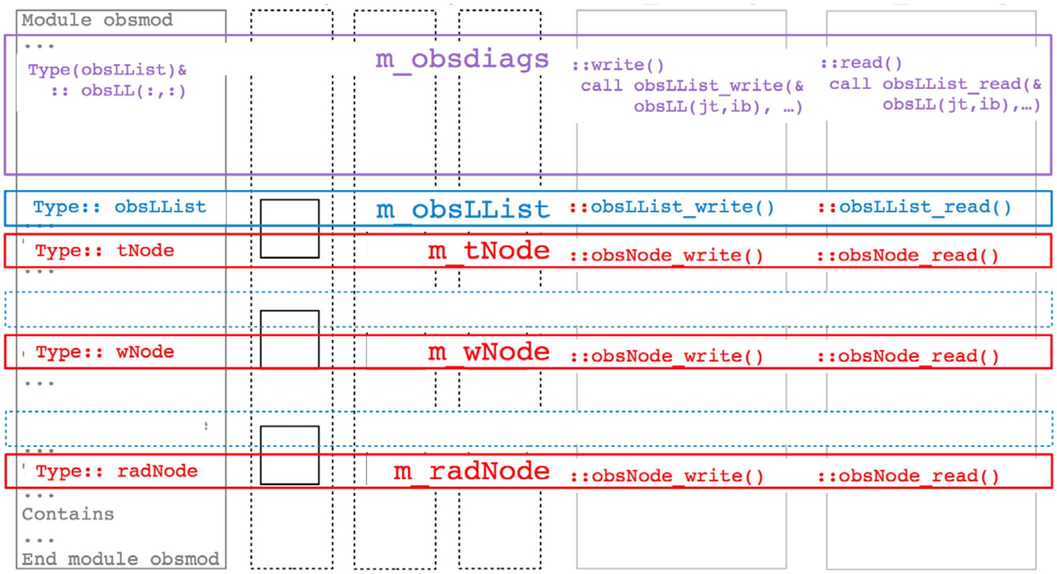


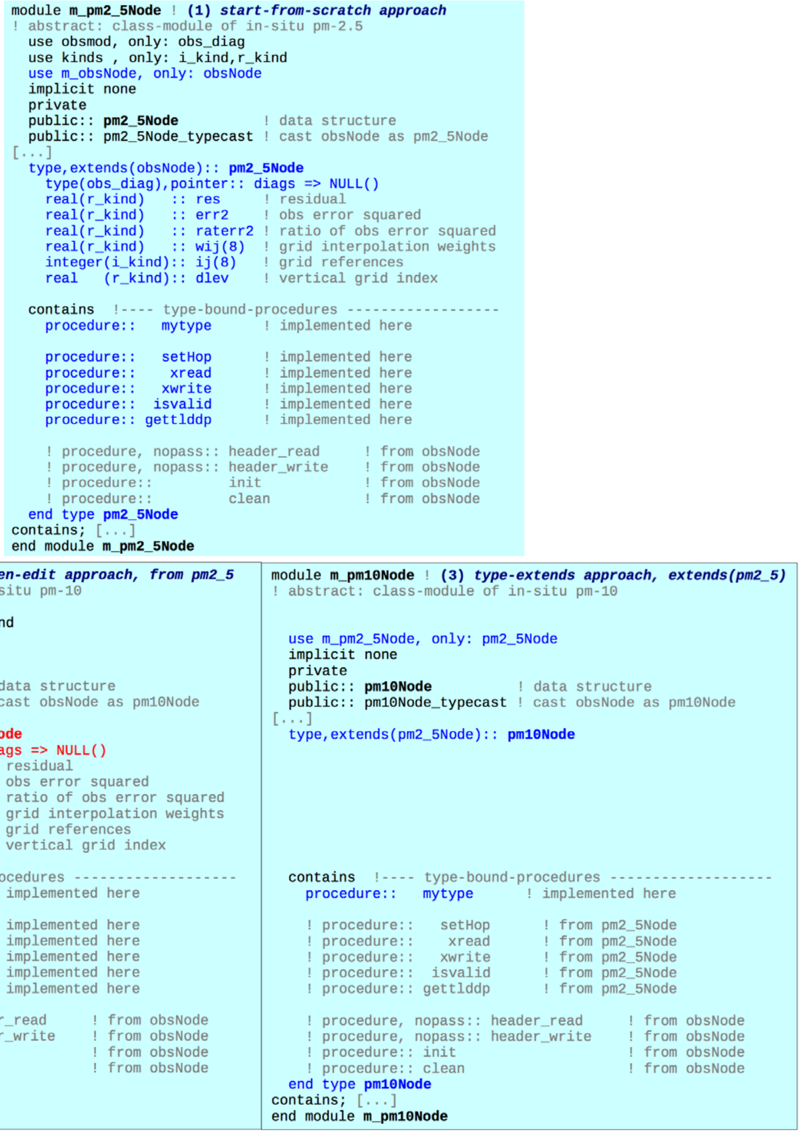
Figure 2. *Schematic view of current* ***polymorphic*** *implementation related to the observation operators in GSI. In this new construct, each given observation type (e.g., the temperature operator tNode) controls its own operations: initialization, finalization, read, write (highlighted in colored horizontal blocks), with additional procedures (setupX, intX, stpX) to be encapsulated soon in the future.*

(intX), their conjugate-gradient step calculations (stpX), and I/O-related procedures are placed independently from the type they relate to, namely, *X_ob_type*.

In an OO world, these procedures should be thought of as methods of a particular type (object). Each self-contained type controls its own methods. For example, the nonlinear and linearized operators related to the radiance observation type should be among the methods controlled by the radiance type. There is also an *abstract* observation type that represents any concrete observation type under consideration of a generic algorithm implementation. A generic implementation then dispatches to concrete type-bound-procedures on the fly, at run-time. A schematic illustration of a modular encapsulation of the methods of each type into separate components is seen in Figure 2. Notice the horizontal arrangement of the blocks in this figure compared the vertical arrangement of those in Figure 1.

This particular restructuring of the observation types reveals the similarities among different types. More important, modularization of the types allows for simplification of the writing of many of them by exploiting their similarities and letting the compiler create the contents of similar required types on the fly. This is done in Fortran by defining a type as an extension of another existing type. Figure 3 provides code snippets illustrating this case. Examination of the code reveals that the methods associated with observation types pm2.5 (top) and pm10 (bottom), which are aerosol-like observations related with different particle sizes, are largely the same. Therefore, the type for pm10 (bottom right) can simply extend the type for pm2.5,

(continued on page 12)



```
module m_pm2_5Node ! (1) start-from-scratch approach
! abstract: class-module of in-situ pm-2.5
  use obsmod, only: obs_diag
  use kinds , only: i_kind,r_kind
  use m_obsNode, only: obsNode
  implicit none
  private
  public:: pm2_5Node          ! data structure
  public:: pm2_5Node_typecast ! cast obsNode as pm2_5Node
[...]
  type,extends(obsNode):: pm2_5Node
    type(obs_diag),pointer:: diags => NULL()
    real(r_kind)   :: res     ! residual
    real(r_kind)   :: err2    ! obs error squared
    real(r_kind)   :: raterr2 ! ratio of obs error squared
    real(r_kind)   :: wij(8)  ! grid interpolation weights
    integer(i_kind):: ij(8)   ! grid references
    real   (r_kind):: dlev    ! vertical grid index

  contains  !---- type-bound-procedures ------------------
    procedure::   mytype      ! implemented here

    procedure::   setHop      ! implemented here
    procedure::    xread      ! implemented here
    procedure::   xwrite      ! implemented here
    procedure::  isvalid      ! implemented here
    procedure:: gettlddp      ! implemented here

    ! procedure, nopass:: header_read     ! from obsNode
    ! procedure, nopass:: header_write    ! from obsNode
    ! procedure::         init            ! from obsNode
    ! procedure::         clean           ! from obsNode
  end type pm2_5Node
contains; [...]
end module m_pm2_5Node
```

```
module m_pm10Node ! (2) copy-then-edit approach, from pm2_5
! abstract: class-module of in-situ pm-10
  use obsmod, only: obs_diag
  use kinds , only: i_kind,r_kind
  use m_obsNode, only: obsNode
  implicit none
  private
  public:: pm10Node          ! data structure
  public:: pm10Node_typecast ! cast obsNode as pm10Node
[...]
  type,extends(obsNode):: pm10Node
    type(obs_diag),pointer:: diags => NULL()
    real(r_kind)   :: res     ! residual
    real(r_kind)   :: err2    ! obs error squared
    real(r_kind)   :: raterr2 ! ratio of obs error squared
    real(r_kind)   :: wij(8)  ! grid interpolation weights
    integer(i_kind):: ij(8)   ! grid references
    real   (r_kind):: dlev    ! vertical grid index

  contains  !---- type-bound-procedures ------------------
    procedure::   mytype      ! implemented here

    procedure::   setHop      ! implemented here
    procedure::    xread      ! implemented here
    procedure::   xwrite      ! implemented here
    procedure::  isvalid      ! implemented here
    procedure:: gettlddp      ! implemented here

    ! procedure, nopass:: header_read     ! from obsNode
    ! procedure, nopass:: header_write    ! from obsNode
    ! procedure:: init                    ! from obsNode
    ! procedure:: clean                   ! from obsNode
  end type pm10Node
contains; [...]
end module m_pm10Node
```

```
module m_pm10Node ! (3) type-extends approach, extends(pm2_5)
! abstract: class-module of in-situ pm-10


  use m_pm2_5Node, only: pm2_5Node
  implicit none
  private
  public:: pm10Node          ! data structure
  public:: pm10Node_typecast ! cast obsNode as pm10Node
[...]
  type,extends(pm2_5Node):: pm10Node








  contains  !---- type-bound-procedures ------------------
    procedure::   mytype      ! implemented here

    ! procedure::   setHop      ! from pm2_5Node
    ! procedure::    xread      ! from pm2_5Node
    ! procedure::   xwrite      ! from pm2_5Node
    ! procedure::  isvalid      ! from pm2_5Node
    ! procedure:: gettlddp      ! from pm2_5Node

    ! procedure, nopass:: header_read     ! from obsNode
    ! procedure, nopass:: header_write    ! from obsNode
    ! procedure:: init                    ! from obsNode
    ! procedure:: clean                   ! from obsNode
  end type pm10Node
contains; [...]
end module m_pm10Node
```

Figure 3. *Within OO concepts, one can recognize similarities among different components, thus allowing for code reuse, improvement for readability, and easier management of differences for* ***extensibility****. As an example, the figure compares the type controlling observations from the abstract to pm2.5 (top), and then to pm10 (bottom). Blue-colored headers of three panels highlight different approaches to create a new type. Exploitation of their similarities means simplification of, say, pm10 by turning it into an extension of pm2.5 (bottom right); in other words, using pm2.5 as a template created on the fly by the compiler.*

with minor differences between the two being accommodated by creating exceptions within the context of pm10 (not shown).

Interface to Guess Interpolation

Use of polymorphism concepts can also be applied to design a flexible (general) handling of the guess (background) interpolations required by the observation simulator (part of the FOO that converts the background to the observable). Just as in the rewrite of the observation operators in modular OO-like framework, the guess interpolations can be handled (1) by development of an interface layer to virtually support generic (abstract) GSI guess state interpolation applications, and (2) by recognizing that every interpolator can be implemented as an extension of corresponding abstract interpolators.

At this stage, the discussion that follows is part of a prototype under test at NASA's Global

(continued on page 13)

Modeling and Assimilation Office (NASA/GMAO). The snippets of code shown below are not final, and consideration is taking place with feedback from other members of the JEDI team to establish consensus on the approach presented here. Nevertheless, discussions have been rather positive and favorable toward what follows, so we feel confident that only minor adjustments will be necessary.

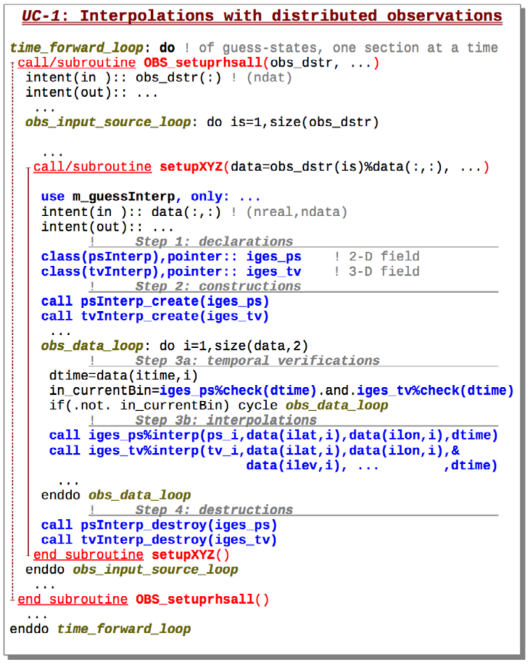
A key component to the present design of an abstract interface is the realization that things like the grid, resolution, partition, choice of variables, and specific partition information related to the state vector should not appear in the interface. This makes the interface rather generic. The only things exchanged between the calling programs and the underlying interfaces are physical quantities, such as variables, 2D or 3D spatial locations, time of observations, and in certain cases, the processor identifiers. A code snippet of possible calls from within a general (setupXYZ) nonlinear GSI operator is given in Figure 4. The abstraction layer is designed to honor all current GSI use-cases and functionalities. Very few assumptions are needed about the guess-state for the interpolations, so the approach here leaves sufficient room for expansion of functionality.

User-Specific Subcomponents

There are multiple examples in GSI where its internal components may be specific to

(continued on page 14)

Figure 4. *With abstraction **m_guessInterp**, (UC-1; left) is a GSI use-case of guess interpolators with distributed observations, while (UC-2; right) is a GSI use-case of inquiring for processor destinations with respect to guess interpolators. Both use-cases are schematically shown in the code snippets of planned implementations. These are very close to the current GSI implementation, but generic. Blue-colored statements highlight complete life cycles of individual interpolators created by m_guessInterp, with few assumptions about concrete implementation of specific interpolators.*

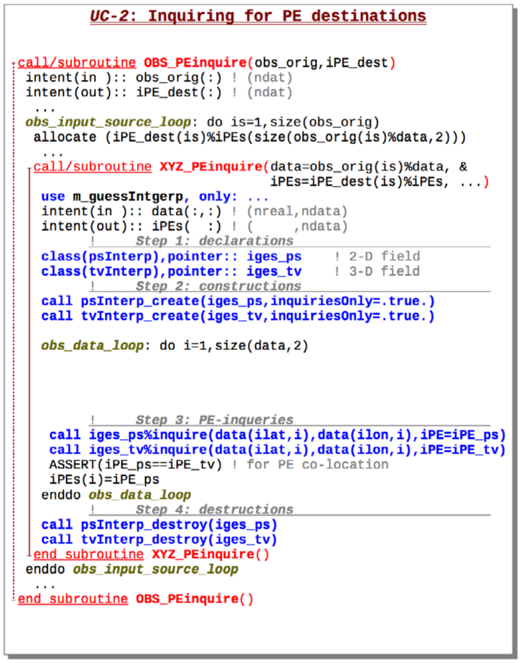


```
       UC-1: Interpolations with distributed observations

time_forward_loop: do ! of guess-states, one section at a time
 call/subroutine OBS_setuprhsall(obs_dstr, ...)
  intent(in ):: obs_dstr(:) ! (ndat)
  intent(out):: ...
   ...
  obs_input_source_loop: do is=1,size(obs_dstr)


    ...
   call/subroutine setupXYZ(data=obs_dstr(is)%data(:,:), ...)

    use m_guessInterp, only: ...
    intent(in ):: data(:,:) ! (nreal,ndata)
    intent(out):: ...
          !      Step 1: declarations
    class(psInterp),pointer:: iges_ps    ! 2-D field
    class(tvInterp),pointer:: iges_tv    ! 3-D field
          !      Step 2: constructions
    call psInterp_create(iges_ps)
    call tvInterp_create(iges_tv)
     ...
    obs_data_loop: do i=1,size(data,2)
          !      Step 3a: temporal verifications
     dtime=data(itime,i)
     in_currentBin=iges_ps%check(dtime).and.iges_tv%check(dtime)
     if(.not. in_currentBin) cycle obs_data_loop
          !      Step 3b: interpolations
     call iges_ps%interp(ps_i,data(ilat,i),data(ilon,i),dtime)
     call iges_tv%interp(tv_i,data(ilat,i),data(ilon,i),&
                              data(ilev,i), ...        ,dtime)
      ...
    enddo obs_data_loop
          !      Step 4: destructions
    call psInterp_destroy(iges_ps)
    call tvInterp_destroy(iges_tv)
   end subroutine setupXYZ()
  enddo obs_input_source_loop
   ...
 end subroutine OBS_setuprhsall()
  ...
enddo time_forward_loop
```



```
            UC-2: Inquiring for PE destinations


 call/subroutine OBS_PEinquire(obs_orig,iPE_dest)
  intent(in ):: obs_orig(:) ! (ndat)
  intent(out):: iPE_dest(:) ! (ndat)
   ...
  obs_input_source_loop: do is=1,size(obs_orig)
   allocate (iPE_dest(is)%iPEs(size(obs_orig(is)%data,2)))
    ...
   call/subroutine XYZ_PEinquire(data=obs_orig(is)%data, &
                                 iPEs=iPE_dest(is)%iPEs, ...)
    use m_guessIntgerp, only: ...
    intent(in ):: data(:,:) ! (nreal,ndata)
    intent(out):: iPEs(  :) ! (      ,ndata)
          !      Step 1: declarations
    class(psInterp),pointer:: iges_ps    ! 2-D field
    class(tvInterp),pointer:: iges_tv    ! 3-D field
          !      Step 2: constructions
    call psInterp_create(iges_ps,inquiriesOnly=.true.)
    call tvInterp_create(iges_tv,inquiriesOnly=.true.)

    obs_data_loop: do i=1,size(data,2)




          !      Step 3: PE-inqueries
     call iges_ps%inquire(data(ilat,i),data(ilon,i),iPE=iPE_ps)
     call iges_tv%inquire(data(ilat,i),data(ilon,i),iPE=iPE_tv)
     ASSERT(iPE_ps==iPE_tv) ! for PE co-location
     iPEs(i)=iPE_ps
    enddo obs_data_loop
          !      Step 4: destructions
    call psInterp_destroy(iges_ps)
    call tvInterp_destroy(iges_tv)
   end subroutine XYZ_PEinquire()
  enddo obs_input_source_loop
   ...
 end subroutine OBS_PEinquire()
```

a particular application. One of the simplest examples is the placement of timers. A particular user, or group of users, might prefer using their own timer utility library to assess performance of internal subcomponents of the code. GSI is presently enabled with calls to start, end, and summarize timings obtained by user-provided timing library. In its default use by the National Centers for Environmental Prediction Environmental Modeling Center (NCEP/EMC), a timing library is not typically provided when the executable of GSI is created; others, such as NASA/GMAO, provide a library that automatically supplies timing information after each execution of GSI. To avoid unnecessary code localization, presently, a default do-nothing stub timer is provided in GSI through a set of Fortran implicit interfaces, so that without GSI code changes, each user application can choose to swap the stub with a functioning timer during the linkage stage.

The same concept is also used by several GSI stubs as a crude approach of managing complex user-specific extensions, including a stub of tangent-linear and adjoint models for 4DVar, and a stub of ensemble background fields, etc. With true polymorphism, stub-swapping by users can be avoided by bringing minor enhancements to the current stub mechanism, where user-specified stub replacements can be implemented and configured as formal extensions.

In Summary

The bottom-up, initial steps toward refactoring of GSI discussed above are expected to facilitate maintainability, extensibility, and scalability of the software. The approach turns GSI into a code that is as object-oriented as any Fortran code can be at this stage in the language support to OO concepts. Abstraction and encapsulation are expected to be the main contributors to enhanced maintainability. The polymorphic implementation of some of its main functionalities will contribute to facilitate extensibility. An easier-to-maintain and easier-to-extend software will facilitate identification of computational bottlenecks and consequent improvement in software scalability.

Acknowledgements
We would like to thank the GSI Committee for supporting the introduction of the concepts discussed here in the GSI software. We thank Thomas Auligné for his energetic enthusiasm to refactor GSI using modern software concepts. And we thank Yannick Trémolet for discussions related to the implementation of a generalized interface to guess-interpolators.

Jing Guo[1] and Ricardo Todling, NASA Global Modeling and Assimilation Office

[1] Additional Affiliation: Science Systems and Applications, Inc., Lanham, MD

DEVELOPMENT EFFORTS

CRTM User and Developer Workshop

The First Community Radiative Transfer Model (CRTM) User and Developer Workshop took place May 16 at the National Center for Weather and Climate Prediction Center (NCWCP) in College Park, MD. This half-day event was intended as the first in an annual series of workshops on all things CRTM.

The workshop was divided into two sections, covering both use and development of CRTM. In the user section, 27 in-person participants came from a broad range of academia, operational centers, and federal research facilities all around the country. An additional 10 people participated online. The following topics were introduced through a series of 20- to 25-minute presentations/tutorials:

(1) Using CRTM as a Stand-alone Radiative Transfer Model (Tong Zhu, CIRA at NOAA/JCSDA)

(2) Using CRTM in GSI (Emily Liu, SRG at NOAA EMC)

(3) CRTM Microwave Applications (Ben Johnson, UCAR at NOAA/JCSDA)

(4) CRTM Infrared Applications (Andrew Collard, IMSG at NOAA EMC)

The second half of the workshop was aimed at developers of CRTM, with a specific focus on improving its underlying scientific and technical basis as well as providing background on tools used to develop and release CRTM. This part of the workshop was attended by 40 people, with approximately 10 online participants.

(1) CRTM Radiative Transfer Overview (Mark Liu, NOAA STAR)

(2) Identifying Coding Standards, Including TL/AD (Open Discussion)

(3) CRTM Optimization (James Rosinski, NOAA ESRL)

(4) Spectral and Transmittance Coefficient Generation (Yong Chen, NOAA STAR)

(5) Future Development and Workshop Wrap-up (Ben Johnson, UCAR at NOAA/JCSDA)

During the workshop, we identified several key goals for the future of CRTM: focusing on improving scientific accuracy and capabilities, performance optimizations, the development of user and developer toolsets, documentation and tutorial creation, and improved developer interaction with the CRTM team.

For the next CRTM User Developer Workshop, based on feedback received during this session, we plan to create a series of

(continued on page 16)

hands-on tutorials that let attendees work through common use cases and allow developers to work with experts to learn how to modify and update CRTM in a consistent fashion. By this time, we also expect to have released a series of toolsets that enable users and developers to get the most out of CRTM.

The CRTM team and the JCSDA would like to thank all the participants and tutors who made this workshop a success. If you'd like to keep up with regular developments of the CRTM, email benjamin.t.johnson@noaa.gov to can join the bi-weekly CRTM teleconference, where we discuss a wide range of scientific and technical issues.

Benjamin Johnson, University Corporation for Atmospheric Research (UCAR) at NOAA/JCSDA

Meeting Summary: 15th Annual JCSDA Science Workshop

The 15th Annual JCSDA Science Workshop was held May 17-19, 2017, in College Park, MD, at the NOAA Center for Weather and Climate Prediction. This event provided a forum for scientists associated with all of the JCSDA partner institutions and the external research community to share some of the latest developments in satellite data assimilation, particularly as these relate to the status of JCSDA priorities and projects. More than 70 registered participants contributed to the three-day event composed of eight oral sessions, two poster sessions, and a multitude of informal discussions. Numerous local "walk-ins" took advantage of the opportunity to participate as well.

Representing the Management Oversight Board (MOB), Dr. Steven Pawson of NASA opened the workshop on Wednesday morning, emphasizing the MOB's confidence that the new JCSDA management structure will lead to more effective collaboration—and progress. The JCSDA Director, Dr. Thomas

(continued on page 17)

Photographed by S. Dutta

Aulignė, then proceeded to describe that structure in detail and explain how JCSDA projects are organized and executed within it. To conclude the morning sessions, members of the JCSDA Executive Team provided in-depth presentations on the status of contributions from the individual JCSDA partner institutions toward state-of-the-art satellite data assimilation.

Wednesday afternoon featured two oral sessions, the first consisting of four presentations on atmospheric composition and assimilation of aerosol data. The second session was devoted to improvements of data assimilation systems, emphasizing common formats and infrastructure, and highlighted by an overview of the newly established Joint Environment for Data-assimilation Integration (JEDI) project. A lively poster session followed, with most participants lingering until supper inevitably called.

Oral sessions on Thursday morning featured the Community Radiative Transfer Model (CRTM), including a project management summary and talks devoted to progress with specific science (microwave scattering for snow and graupel; supporting specific sensor calibration and validation) and/or computational improvements for accuracy and speed. This provided a logical lead-in to the first afternoon session devoted to sensor-specific DA, with emphasis on using microwave sounding data in operational hurricane and mesoscale modeling systems, as well as to the assimilation of GPSRO data from a growing variety of sources including COSMIC-2 and commercial pilot weather-data observations. Two shorter modules addressing land surface and ocean DA paved the way to another spirited poster session to conclude the day.

Friday morning opened with a series of talks on observing system assessment and optimization, covering OSEs, OSSEs, and comparison of multiple FSOI assessments. The final session featured three presentations addressing the assimilation of observations impacted by clouds and precipitation. This led to a lively discussion that cut into the time for open discussion; thus, the formal proceedings were closed with remarks from the NESDIS/STAR Director, Harry Cicanek, who encouraged the participants to build on the work already done and take advantage of the new project management to accelerate exploitation of satellite data in the nation's operational forecast systems.

All oral presentations are available on the JCSDA website, located at:
http://www.jcsda.noaa.gov/meetings_Wkshp2017_agenda.php

The poster presentations may be accessed online at:
http://www.jcsda.noaa.gov/meetings_Wkshp2017_posters.php

We hope that all participants have a very exciting and productive year, and we look forward to next year's workshop in 2018!

Jim Yoe, JCSDA

Unsolicited articles for the JCSDA Quarterly Newsletter are encouraged as are suggestions for seminar speakers or topics. Please send them to Biljana Orescanin, biljana.orescanin@noaa.gov.

PEOPLE

Welcome Dr. Guillaume Vernieres

Dr. Guillaume Vernieres joined the National Oceanic and Atmospheric Administration's Environmental Modeling Center (NOAA/EMC) in April in support of JCSDA to implement sea-ice/ocean data assimilation capability within the Joint Effort for Data-assimilation Integration (JEDI) project. While the short-term objective of this project is to implement sea-ice data assimilation within an early JEDI prototype and Gridpoint Statistical Interpolation (GSI), the long-term goal is to integrate current state-of-the-art ocean and sea-ice data assimilation system within the JEDI.

Guillaume earned a Ph.D. in Physical Oceanography with a minor in Applied Mathematics at Oregon State University in 2006. His graduate studies focused on weak and strong constraint 4DVAR data assimilation for the ocean.

After graduating, Guillaume joined the Statistical and Applied Mathematical Institute (SAMSI) and the Applied Mathematics department at the University of North Carolina at Chapel Hill as a postdoctoral fellow. During his fellowship, he participated in the development of parameter estimation methods for models of cerebral blood flow and worked on the application of ensemble assimilation technique to Ocean Lagrangian observations.

Before joining the JCSDA, Guillaume spent several years at NASA Goddard Space Flight Center as a research scientist and contributed to the Global Modeling and Assimilation Office (GMAO) effort on the development and implementation of ocean and sea-ice data assimilation and seasonal prediction capability. His research interests ranged from the development of machine learning algorithms for the bias correction of salinity retrieval from L-band radiometer to covariance modeling for the assimilation of altimeter and hydrographic measurements. Guillaume was a lead contributor to MERRA-Ocean, an ocean-sea ice retrospective analysis used for the initialization of the GMAO seasonal forecast. During his last year at the GMAO, he integrated the University of Maryland Local Ensemble Transform Kalman Filter (UMD LETKF) within the GEOS-5 coupled model for the assimilation of ocean and sea-ice observations. This system will be operational in the summer of 2017.

In his spare time, Guillaume enjoys spending time with his family, sailing and drinking coffee.

CAREER OPPORTUNITIES

Opportunities in support of JCSDA may be found at http://www.jcsda.noaa.gov/careers.php as they become available.

SCIENCE CALENDAR

UPCOMING EVENTS

MEETINGS AND EVENTS SPONSORED BY JCSDA

DATE	LOCATION	TITLE
11–14 July, 2017	NOAA NCWCP, College Park, MD.	GSI/EnKF Community Tutorial*
TBD 2018	TBD	JCSDA Summer Colloquium on Satellite Data Assimilation
TBD 2018	TBD	JCSDA 16th Technical Review Meeting & Science Workshop on Satellite Data Assimilation

* Note from the GSI/EnKF Tutorial Organizing Committee: This tutorial will be a three and a half-day event. Lectures and hands-on sessions will be provided by invited speakers from major GSI and EnKF development teams on July 11-13 (Monday-Thursday), followed by an optional practical session on Friday morning (July 14).

All are invited to the GSI Workshop and the GSI Tutorial lecture portion. However, due to the constraints of physical space and staffing, we can only accommodate a maximum of 40 participants for the tutorial hands-on portion.

Registration for the hands-on portion has closed. Further details and information are available on the DTC website (http://www.dtcenter.org/) and the community GSI website (http://www.dtcenter.org/com-GSI/users/).

JCSDA seminars are generally held on the third Wednesday of each month at the NOAA Center for Weather and Climate Prediction, 5830 University Research Court, College Park, MD. Presentations are posted at http://www.jcsda.noaa.gov/JCSDASeminars.php prior to each seminar. Off-site personnel may view and listen to the seminars via webcast and conference call. Audio recordings of the seminars are posted at the website the day after the seminar. If you would like to present a seminar, contact Ling Liu, ling.liu@noaa.gov, or Biljana Orescanin, biljana.orescanin@noaa.gov.

MEETINGS OF INTEREST

DATE	LOCATION	WEBSITE	TITLE
31 July–2 August 2017	Vancouver, Canada	http://www.cs.ubc.ca/~greif/precon17/	Preconditioning 2017: International conference on preconditioning techniques for scientific and industrial applications
11–15 September 2017	Florianopolis, Brazil	http://www.cptec.inpe.br/das2017/	Seventh International WMO Symposium on Data Assimilation
October 23-27, 2017	ECMWF (Reading, UK)	https://events.oma.be/indico/event/18/page/7	13th Stratosphere-troposphere Processes And their Role in Climate (SPARC) Data Assimilation workshop
29 November–5 December, 2017	Darmstadt, Germany	https://cimss.ssec.wisc.edu/itwg/itsc/itsc21/index.html	21st International TOVS Study Conference
11–15 December 2017	New Orleans, USA	http://fallmeeting.agu.org/2016/future-meetings/	American Geophysical Union Fall Meeting
7–11 January 2018	Austin, TX	https://annual.ametsoc.org/2018/	98th AMS Annual Meeting