Explanation of Software for Generating Simulated Observations For the GMAO OSSE prototype

By
Ronald M. Errico (GMAO and GEST)
Runhua Yang (GMAO and SSAI)
Jing Guo (GMAO and SAIC)

20 August 2008

Acknowledgements

We are greatly appreciative of several people: The idea of using an artificially elevated surface to introduce the effects of clouds or surface emissivity errors was suggested by Joanna Joiner. Reading and writing of BUFR data files was assisted by Meta Sienkiewicz. Hui-Chun.Liu provided assistance reading and writing AIRS data. Both she and Tong Zhu (NCEP) assisted with use of the NCEP Community Radiative Transfer Model. Ricardo Todling and Ronald Gelaro helped our use of the NCEP/GMAO GSI data assimilation system software, especially its adjoint version used to expedite tuning. Additional software was provided by Arlindo da Silva and Ravi.Govindaraju.

1. Introduction

In order to understand the design and function of the present code to generate simulated observations for the prototype GMAO OSSE, it is necessary to understand our goal. This is to:

> *Quickly generate a prototype baseline set of simulated observations that is significantly "more realistic" than the set of baseline observations used for the previous NCEP/ECMWF OSSE.*

By *quickly* here we mean within 9 months from the inception of the work (in December 2007), if possible. This seemed a reasonable goal if we obtained sufficient cooperation from others and if no dramatic unforeseen obstacle presented itself. An example of the latter would be if we discovered that, although the clouds provided by the nature run appeared to have realistic seasonal and zonal means, their distributions at individual times for effects on satellite observed radiances were fatally unrealistic (We do not expect such a result, but some other unexpected, equivalently fatal flaw in our approach can still be encountered). Or, if we need to research many required details ourselves without relying on expertise present, further unnecessary delay can occur. Presently, however, we believe our 9-month goal is achievable.

The word *prototype* signals our intention to develop an even more realistic and complete dataset in the future. We know how to do better regarding several aspects of the simulations and we know which observations have so far been neglected. Several of these aspects and all these observations will be mentioned in what follows. Their present omissions are simply due to time. Some missing aspects are expected to have negligible impact on the realism of the observations. Most actually concern realism of treatments of errors in the observations rather than their information content, as will be explained in a later section. The missing observations, except for MSU, have been shown to have negligible impacts within the present GMAO/NCEP data assimilation system according to the metrics we will be employing for OSSE validation.

Baseline refers to the set of observations that were operationally utilized by the GMAO DAS during 2005. This set should be similar, but not identical, to the set used by NCEP during that period. It is this entire set that will eventually be included in the OSSE validation studies, although for expediency in developing the prototype, some lesser observations have been initially neglected.

There is necessarily a tradeoff between the intentions of the subjective words *quickly* and *significantly*. Plans of precisely what and when a particular development occur will change as we better assess the time required and the benefits expected. As a first measure of improvement, however, we have something quite specific in mind. This concerns comparisons of temporal variances of analysis increments produced by the DAS for baseline real and OSSE assimilations. This specific goal is described in section 2.

In order that the baseline OSSE adequately validates, it is beneficial if the observation simulation procedure is tunable in several ways. Since different models, grid resolutions, and grid structures are used to produce the nature run and DAS, some representativeness error is already implicitly included in the simulated observations before any explicit error is added. How much implicit error is present is unclear, however, and therefore some tuning of the explicit error to be added is required. Also, it is unclear how well the cloud information produced by the nature run is realistic regarding those aspects that impact radiance transmissions through the atmosphere at observation times (All we have seen thus far are validations of time and zonal mean cloud information from the nature run). So, having tunable parameters that will permit easy compensation for possible deficiencies in the nature run clouds is beneficial.

When we first began this project, we expected other investigators to produce simulations of most types of baseline observations. So, for example, we originally committed to only produce simulated IR radiances for HIRS2/3 and AIRS. As we proceeded, however, we realized that little additional work would be required to also produce AMSU-A/B simulations and even observations for conventional observations. Simulations for all observations and their corresponding errors use a common set of basic software. There is therefore no need for us at the GMAO to use the cumbersome, multiple step, data exchange process with NCEP that we initially were utilizing.

2. The former NCEP/ECMWF OSSE

Our familiarity with the former NCEP/ECMWF OSSE is limited to the work involving M. Masutani, most of which is unpublished. This specifically refers to work using a ECMWF model from the 1990s run at T213L31 for the nature run. Only about 5 weeks are simulated. That is a short period to produce statistically significant DAS results. The resolution is also less than that of current operational analysis. None-the-less, these OSSEs were an improvement over past ones because an extensive set of validation experiments were performed by comparing results from corresponding data-denial experiments in the OSSE and real DAS frameworks.

We became involved in the former OSSE due to our interest in using the baseline results to estimate characteristics of analysis error. This motivation and key results are presented in Errico et al. (Meteorologische Zeitschrift December 2007, p 695-708). As part of this study, we also produced some validation measures complimenting those investigated by Masutani and colleagues. Our measures included standard deviations of time and zonal mean variances of analysis increments measured at 1200 UTC each day for the last 21 days of the NCEP baseline assimilation. This measure was produced for both the OSSE and corresponding real analysis frameworks. For both frameworks, two sets of results were produced: one used the full set of observations used operationally during February 1993; the other excluded satellite radiance observations.

A key result from the validation performed by us appears in Figs. 2.1-2.2 here. These show standard deviations of analysis increments (analysis minus background fields) for the eastward component of velocity (u) for 4 experiments. The pair of plots in each figure is for real DAS and corresponding OSSE statistics. Fig. 2.1 considered all "conventional" observations plus satellite tracked winds, but no satellite observed radiances for temperature and moisture information. Fig. 2.2 also included those radiances.

The results in Fig. 2.1 show fairly good comparison especially considering (1) that 3 weeks of analyses provide only a small sample and (2) that, given the nature of chaos, the corresponding real and nature-run fields over that short period may have very different characteristics regarding how they effect errors in the DAS even if the nature run is otherwise totally realistic. In other words, the dynamic instabilities present in the real and simulated datasets may be significantly different just because the synoptic states differ. The results in Fig. 2.2 show that increments are slightly reduced when radiances are used, suggesting that the analysis and corresponding truth are closer to each other when the additional observations are used, as expected. In Fig. 2.2, however, the two plots look less like each other than the two paired in Fig. 2.1. This suggests that perhaps some aspect of the simulation of radiance observations is unrealistic in the OSSE, creating a poorer validation when those observations are used. Unfortunately, it is difficult to make a stronger statement, since the comparison is rendered difficult because these are old plots produced at different times using different color tables, etc.

One known unrealism in the production of simulated radiance observations in the former NCEP/ECMWF OSSE is that the locations of simulated cloud-free radiances was defined as the identical locations of cloud-free radiances as determined by the real DAS quality control procedure in the real assimilation for the corresponding time. Thus, in dynamically active regions where clouds are often present in reality (e.g., in cyclones) the OSSE may have simulated observations although such regions would tend to be less well observed in reality. This may skew the OSSE statistics, because dynamically stable and unstable regions then have equal likelihoods of being well observed. Since we have identified this problem and suspect it may be important, it is one specific improvement being made for the new prototype OSSE at the GMAO.

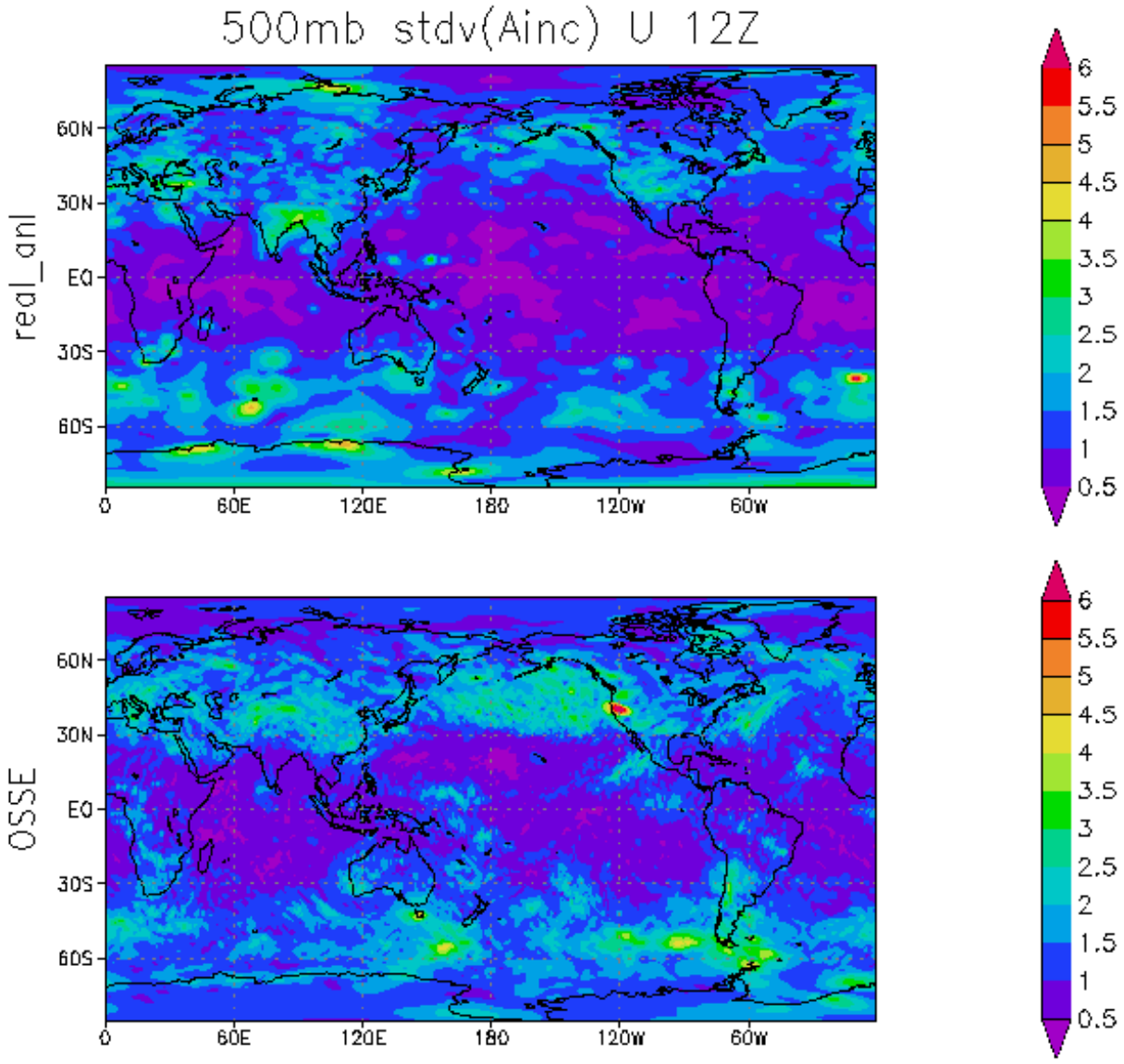

Figure 2.1: Standard deviations of analysis increments of the eastward wind component on the sigma=0.5 surface. The average is over 21 consecutive analyses produced for 12Z during a period in February 1993 for a real analysis (top) and corresponding OSSE (bottom). No satellite radiances or temperature/moisture retrievals were used in either analysis. Units of *u* are m/s.

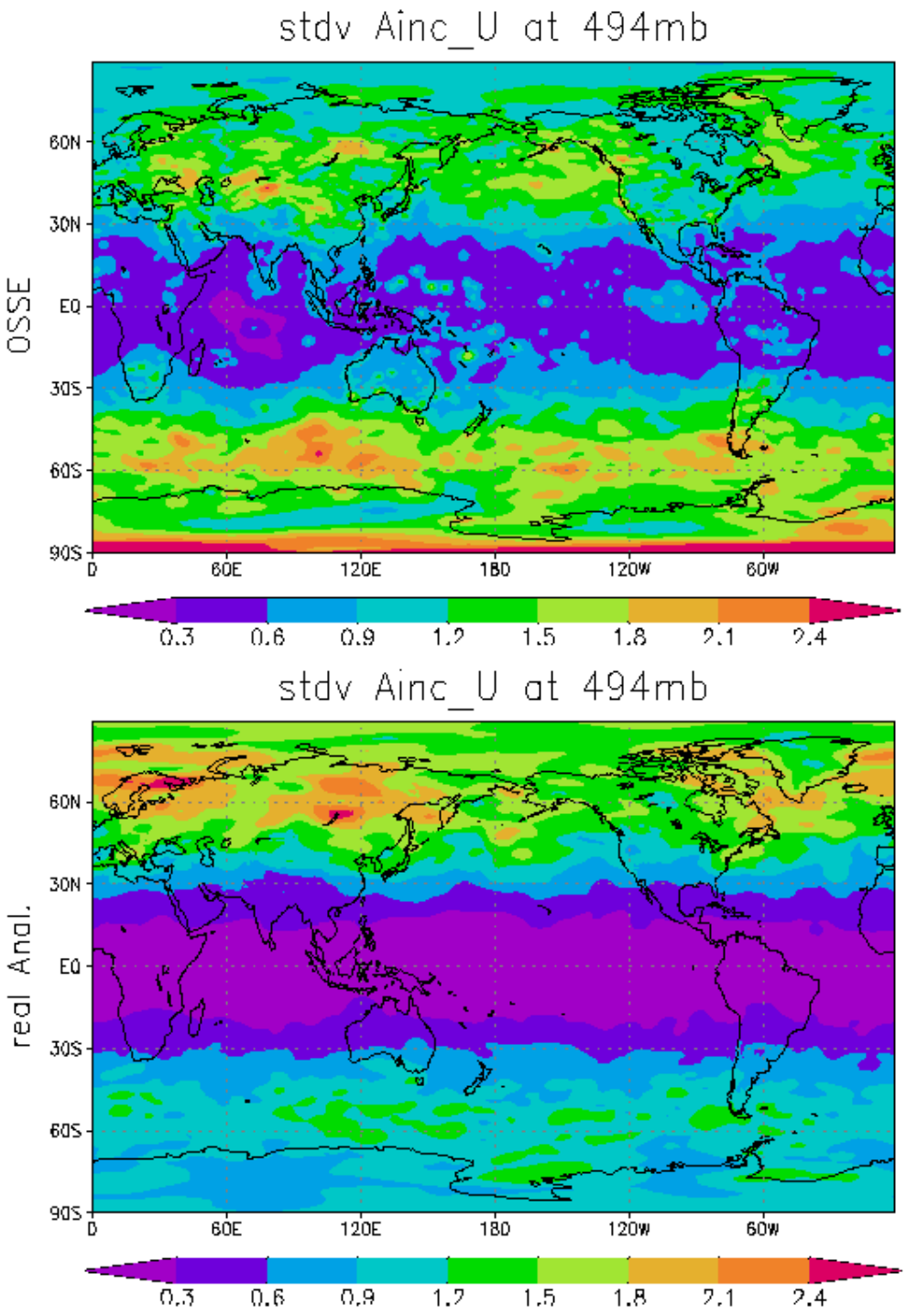

Figure 2.2: Like figure 1, except both the real and OSSE analysis include satellite observed radiances. Note that the OSSE results are now at the top and the color tables, while identical for the pair here, are different than those for the pair in Fig. 1.

3. Basic formulation for version P1

This first prototype (version P1) of the simulated observations includes all observation types assimilated operationally by the GMAO during 2005 except for TPW, GOES precipitation retrievals, GOES-R radiances, and MSU. All but the MSU have been shown to have negligible impact operationally, although that of course may be more a consequence of how they were used by GSI than an indication of the actual quality of the real observations themselves. MSU was omitted by accident, and an attempt to include it will be made as soon as possible.

In order to simulate a realistic number and spatial distribution of observations, the set of real observations archived for the period of the OSSE are uses as a template. These provide observation locations, but not observation values. So, there is no need to use an orbit model for a satellite that was already operationally used at that time. The use of this information is not as simple as it suggests, however, because there are also quality control issues that need to be addressed as described below for individual observation types where appropriate.

For conventional values (i.e., temperature, wind, and specific humidity, but not radiance brightness temperatures) for observations, the GSI reads from a "prepbufr" file that contains only observations that have passed some gross quality control checks. The simulated P1 observations only use observation locations present in this file. Thus, their number has been partially thinned based on the QC conditions that occurred in reality. Additional QC checks occur during execution of GSI. Some tuning of the simulated observation error may be required to get realistic rates of final acceptance (see section 4).

The simulated observations produced are written to a file in BUFR format that is designed to look like the original file that contained the real observations for the corresponding time. If the original file lead with information about the BUFR table, the one for simulated data does also. If the original file lead with some blank reports (e.g., as for HIRS and AMSU data), so does the simulated one. What has been done in general, however, is to write to the new file only the data that is actually read by the GSI. In fact, for the P1 files, this includes only what is read in the current GMAO version of GSI. That version of GSI successfully reads and interprets all the observational data on the simulation files. Some data that is not presently used, however, may be missing from the file. Other data that is not presently used is included on the file, but without knowing how such information is to be used, its simulation may not be testable yet and minimal care has been expended on its creation. Only the data actually used has been checked.

Changes to the files of simulated observations may be required as GSI evolves. The GMAO version of GSI at the end of 2008 should be very similar to the NCEP version of summer 2008. Once this latest version is available to us, we will make sure that the files are readable in this updated version. In the future, perhaps some WMO standard can be applied to writing these files. To see what is currently written to the files, the module containing the BUFR writing software should be examined.

3.1 Consideration of effects of clouds on IR radiances

Transmittance of IR radiation through the atmosphere is strongly affected by clouds. The modeling of scattering, absorption, and transmittance of radiation by clouds is still in its infancy, especially regarding modeling using computational algorithms fast enough to produce the hundreds of millions of observations required for the OSSE in a reasonable time. Even if such algorithms are available, their performance for a wide range of cloud distributions, particularly for optically thin clouds should first be demonstrated. For the next version of the observation simulations we will explore what possible software may exist for this purpose, but in the meantime, for a variety of additional reasons, we will use a simpler approach.

Currently, the GSI only assimilates what it believes to be radiances unaffected by clouds. If clouds are present, they are either negligibly thin or far enough below the region from where the radiation is effectively emitted. For those cloud-affected observations that are not discarded by the GSI quality-control procedure, differences between cloud free and the real cloud-affected transmittance effectively are considered as an error of representativness (i.e., specifically error in the observation operator). Thus, even if an accurate radiative transfer model is used to simulate the effects of clouds on radiance observations from the nature run, most of that extra effort will simply be discarded as the GSI detects large differences with its cloud free calculation from the background. Those observations only weakly affected by clouds will pass the quality checks, affecting the distribution of "errors" in the observations as considered by the GSI.

The effects of a thick cloud can easily be modeled, since in this case it may be considered as a black body. Thus, a thick elevated cloud appears the same as an elevated surface as far as IR is concerned. IR channels that normally peak lower in the atmosphere will therefore appear much colder. Channels that normally peak much above the cloud level will remain unaffected by the "elevated" surface. Thus, in version P1, the effects of clouds on IR radiation are introduced by simply setting the cloud top temperatures to the atmospheric temperatures at their elevations, and informing the radiative transfer model that the surface is at that level. Thus, the gross effects of clouds are modeled without using a radiative transfer model that explicitly considers clouds. The use of this gross modeling is primarily to obtain a realistic count of cloud-free observations, as a function of radiance channel and consistent with the distribution of clouds in the nature run. Effects of thin clouds on the radiances are handled by appropriately tuning the model that adds representativeness plus instrument errors (see section 4).

At this time, the distributions of cloud-related fields provided in the nature run (specifically profiles of liquid and ice water contents and cloud fractions) have not been sufficiently validated regarding their effects on IR radiances, especially in the presence of only thin clouds. Although examination of time and zonal mean fields of some measures of cloud content in the nature run is useful, their agreement with nature does not ensure that realistic cloud effects will be obtained when they are considered by a radiative transfer model that includes them, even if that model is a good one. While we believe that the cloud related fields in the nature run are much more realistic than in the former

NCEP/ECMWF OSSE, we expect that some important aspects may be unrealistic, especially regarding the prevalence of high thin clouds. Also, the nature run fields refer to averages or the centers of roughly 35 km square boxes, but clear holes may be present for some observations to be unaffected.

In order to expedite the development work in the light of all the above reasons, in version P1 we have included a simple tunable scheme to incorporate effects of clouds in the IR simulated observations. This scheme uses a stochastic function to determine whether radiances are cloud affected, where the probability of that being the case is a function of the fractional cloud cover at 3 levels provided by the nature run data set.

Cloud Presence Algorithm

1. Denote $j = 1, 2,$ or 3 for low, medium or high clouds, respectively.
2. For each j determine a probability p_j that a cloud of that type is within the instrument field of view and is significantly affecting some observed radiance channels. p_j is simply defined as a convienient piecewise linear function of the corresponding cloud fraction f_j provided by the nature run:

$$p_j = \begin{cases} 0. & \text{if } f_j \le a_j \\ 0.5(f_j - a_j)/(b_j - a_j) & \text{if } a_j < f_j < b_j \\ 0.5 + 0.5(f_j - b_j)/(c_j - b_j) & \text{if } b_j < f_j < c_j \\ 1. & \text{if } f_j \ge c_j \end{cases}$$

   a, b, c are three tunable parameters, with separate values for distinct j and instrument type. Note that b_j is the value of f_j for which $p_j = 0.5$.
3. For each j, choose a random number $0 \le r_j \le 1$ from a uniform distribution on that interval.
4. A significant cloud is present if $r_j < p_j$
5. Find the largest j for which a significant cloud has been determined to be present. The cloud top pressure is then assigned to be

$$p_c = \sigma_j p_s$$

   where p_s is the surface pressure at that location and σ_j is another tunable parameter.

Figure 3.1: The tunable algorithm for specifying whether a cloud that may effect radiance transmission is in the field of view of a simulated satellite observation.

Three levels of clouds are considered; low, medium, and high (height) clouds. In the nature run data set, these correspond to pressure ranges p > 0.8 ps, 0.45ps <= p <= 0.8ps, and p < 0.45 ps, respectively, where ps is the surface pressure at that location.

In version P1 here, the presence of each type of cloud is determined by the algorithm described in Fig.3.1. This particular form for the probability functions was chosen because it is both simple and tunable. The four tunable parameters, a, b, c, and sigma are specified for each instrument type: An instrument with small viewing footprint has a greater chance of encountering a hole in the clouds than one with a larger footprint. The cloud top pressure is specified as a fraction of surface pressure (sigma = p/ps) so that low clouds can be present below p=500 hPa over high topography, such as over Tibet.

The probability function used by this algorithm is piecewise linear as shown in Fig. 3.2. If the cloud fraction for a particular level is less than or equal to parameter a, then the field of view is defined as free of clouds at that level. If it is greater than or equal to parameter c, then it is definitely cloud contaminated. If neither of these conditions hold, then the probability P of a cloud being present is between 0 and 1, and b is then the value of the cloud fraction associated with a cloud-contamination probability of 0.5. In this case, whether a contaminating cloud is declared present is determined by drawing a random number 0<x<1 from a uniform probability distribution (using a standard FORTRAN call to random number generator) and comparing it with P: If x<P, then such a cloud is present; otherwise not. The statistics of this procedure are that, e.g., 20% of all the cases when P=0.20 are expected to be declared as cloud affected.

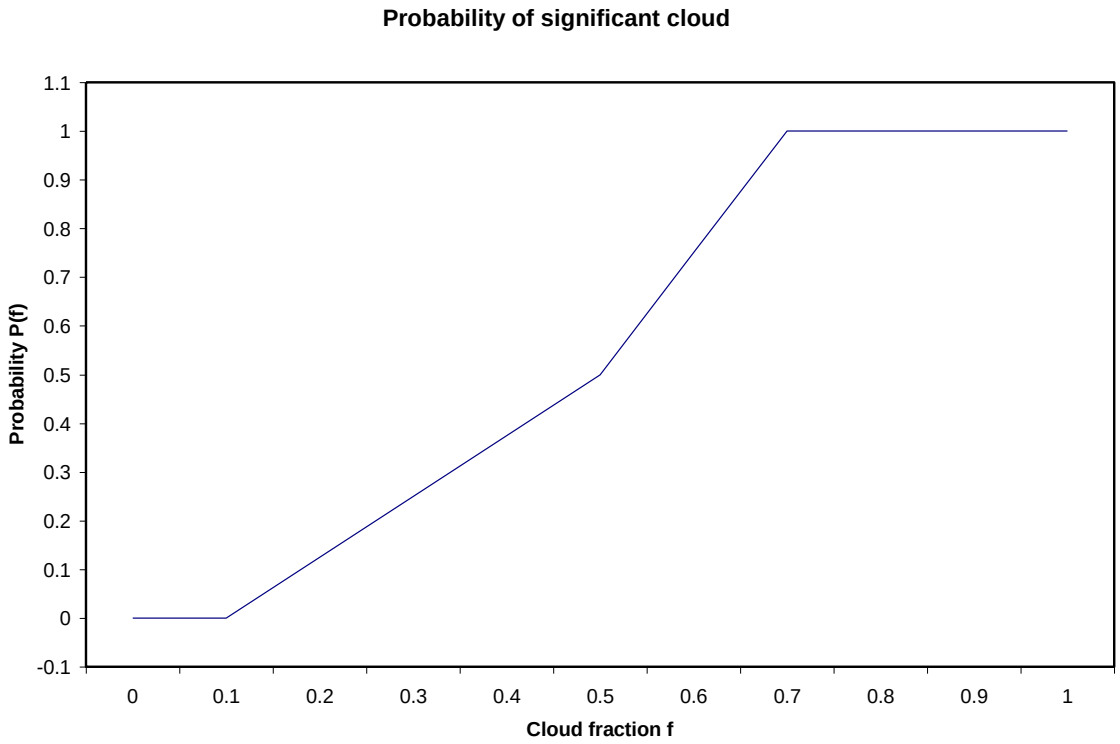

Figure 3.2: Graph of the probability of a significant cloud being in the field of view given the cloud fraction for tuning parameters a=0.1, b=0.5, and c=0.7.

What matters is where the cloud tops are, so first this procedure is done for high clouds, then for middle, and last for low. If a cloud is declared, then sigma is specified as given in the table for that level cloud and clouds at lower levels are not considered. If no radiatively significant clouds are declared present, sigma=1 is specified, indicating that the effective radiative surface is the true surface.

In this procedure there are 12 parameters that can be adjusted. Since we as yet have little experience with tuning these, we offer no guidance at this time. We have tried varying them, however, to see what impact they have on data quality control in GSI and their effects appear to behave as designed. The question of whether this tuning will be sufficient to obtain the degree of validation that we hope is yet to be answered.

3.2 Consideration of MW radiance

In version P1, we are assuming that there are no effects of clouds on microwave transmittance through the atmosphere. We assume emissivity modeling over land and ice due to imperfectly known and highly variable surface conditions is such that surface affected channels have large errors that will lead to observation rejection by the GSI quality control. We further assume that, given the broadness of microwave radiative structure functions, many channels are so affected. Consequently, in version P1, observations over land or ice are computed for a surface elevated to some value of sigma such as 0.7 so that few surface-affected channels are used except over water .

Precipitation is assumed to affect MW radiances. So, we apply a tunable probability model analogous to the one used for modeling cloud effects on IR radiances. Instead of cloud fractions, however, this is a function of the stratiform and convective precipitation rates. Separate probabilities are computed for each, with the effective sigma level for stratiform below that for convection, so that the latter is considered first.

3.3 Biases in radiances

There are several sources of biases in real radiance observations from satellites. Some of these concern the instruments; e.g. the satellite antennas detecting interference from the satellite platform itself. This is inferred especially from an asymmetric scan angle bias, since depending on which way the antenna is pointed, it “sees” a different portion of the platform. Biases can also result from the forward model. They may be difficult to determine if, although systematic, they depend on the synoptic state they are observing. In general, these biases have been estimated to be rather large. They must therefore be removed prior to a data assimilation system attempting to extract the useable information in a standard variational procedure that assumes bias-free observations.

For the version P1 simulated observations that are intended for ingestion in GSI, the sources for creating basis mentioned above are absent. There is no simulated satellite

platform and the forward model (the CRTM) is at most a different version of the same algorithm and program as used in the GSI. Thus, there is no substantial bias introduced and thus there is no need to use radiance bias correction in the GSI when assimilating the P1 observations. By turning off the GSI bias radiance correction, there is no need to spin up files of bias correction coefficients.

Obviously, one source of likely error in the OSSE that is unrealistically absent is remaining significant error in the radiance bias correction algorithm. This can be included by adding some sort of bias to the observations that may have any characteristics an experimenter cares to incorporate. For example, biases that are derived from the GSI bias correction model and coefficients can be added easily. Presumably, however, these would then also be effectively removed by the bias correction. There therefore seems little point in adding such biases unless an experimenter has a specific test of the bias correction in mind. In that case, the biases that are added should be carefully designed to test the specific hypothesis proposed; e.g., effects of biases not described by the GSI algorithm. So biases can be added, but in general there seems to be no need to do so for most experiments.

3.4 Thinning of Radiance Data

Within a data assimilation program, each satellite observation requires a call to a radiative transfer model that can be computationally expensive. Also, geographically close observations can worsen the conditioning of a minimization problem solved by the data assimilation algorithm, thereby increasing computational requirements. For these reasons, GSI therefore use only a small fraction of the satellite observations available to it. It performs a data selection process to choose observations that are well separated in space or time and that are estimated to have the best quality in some sense. This selection process is called observation thinning.

If we produced simulations for all the radiance observations available, most of that effort will be wasted as the observations are thinned by GSI. Also, the computational expense would be great indeed, since each observation would require a call to a radiative transfer model. Therefore, we also thin the data. The procedure is similar to that used by GSI but our thinning is to a lesser degree. In this way we allow the GSI to conduct its own data selection, albeit with a reduced set of observations to consider.

The thinning is conducted by defining "thinning boxes" on the globe. These are approximate squares covering the globe, with their size determined by a user-specified length of their sides. The particular box within which each observation is located is first determined. If it is the first observation considered within that box, it is "placed in the box." If a previously considered observation has been placed in that box, then a selection between the already present and new observation is made. The observation retained is the one less affected by clouds, precipitation, or surface emissivity, as designated by a larger value of its assigned sigma produced by the cloud specification algorithm (see

sections 3.1, 3.2). If two observations have the same value of sigma, then the one closest to the synoptic (central) time being considered is retained. Each thinning box thus contains at most 1 observation.

Only the locations and times of the thinned set of observations are passed to the interpolation software for constructing simulated atmospheric profiles from the nature run. And only those profiles are submitted to the radiative transfer model for creation of simulated radiance observations.

The size of the thinning boxes are user-specified in a resource file (see section 7.1). If the length of a side of one of these boxes is specified as d kilometers, then the number of thinning boxes is approximately m= 5.1 x 10**8/ d**, where the number shown is the earth's area in squared kilometers. If the swaths of a particular satellite cover only a fraction c of the area of the earth, then approximately that fraction of boxes should contain observations, and roughly n=m*c locations will be used to simulate observations by that satellite.

3.5 Rawindsondes

In order to expedite development of the observation simulation software, some liberties were taken with simulating rawindsonde observations in version P1

i. The presence of significant levels is specified by the reported pressure levels of the corresponding real observations. So, the simulated observations may have a high density of observations over some ranges in the vertical, but these are likely not ranges where the variations (e.g., wind shears or temperature inversions) in the nature run fields are especially significant. Likewise, at rawindsonde locations where significant levels occur in the nature run, simulated rawindsonde reports may not include them.

ii. The locations in time and space for all the observations identified during a balloon ascent are specified as being identical. Specifically, the locations and times of all observations for a single balloon ascent are specified as those of the launch station and time. At this stage in the OSSE development, this simplification should not have significant effects on the OSSE validation. The only way in which effects of this simplification can be amplified is if by collocating the observations in latitude and longitude significantly worsens the condition number of the GSI minimization algorithm. At this time, GSI uses the reported wind values rather than determining the winds from the changes in balloon location . Therefore this simplification is expected to have only a small impact.

These shortcomings will be corrected in the next version of the software. This will require "flying" a simulated balloon within the nature run fields. Although some such software has already been developed, being used at NCEP, we suspect that it does not

address the primary deficiency of our P1 rawindsondes that define significant levels based on corresponding real soundings rather than on the nature run fields.

3.6 Wind profiler observations

In the version of GSI currently used at the GMAO, the elevation at which a reported wind observed by a wind profiler is considered to be specified as the pressure level associated with that value. In newer versions of GSI, including that used now at NCEP, the elevation used is the recorded height. In version P1, however, the observed wind values are determined by vertically interpolating from the nature run fields defined on its grid surfaces to the pressure level provided in a report. The corresponding height of the observation in the simulated report is simply copied from the corresponding real observation. Since presumably the real and nature run atmospheric surface pressure and thermal structures differ at any time, although the real observation may have recorded pressures and heights that correspond to the same elevation, that may not be true for the P1 observations.

The GSI will interpret this discrepancy in the P1 profiler observations as an additional source of error, effectively assigning the observations to the wrong elevations. Whether this is a big effect or not, we do not yet know. Correction of this discrepancy eventually will be necessary.

3.7 Cloud-track winds

Wind reports based on tracking clouds or water vapor imaging, in reality, depend on the presence of trackable features. In version P1, the locations of such reports are explicitly those where corresponding real observations were. These locations are not based on the presence of trackable features in the nature run. A P1 observation of cloud track winds may be in a location devoid of clouds in the nature run. This deficiency will be addressed in a later version of the simulated observations. As for the IR observations, this may require a tunable scheme to allow adjustment for possible deficiencies in the NR cloud distribution at instants of time.

3.8 Surface Winds

Values for simulated surface wind observations, either for station reports or scatterometer retrievals, are currently inferred simply by horizontal interpolation of 10 m winds provided in the nature run data set. These are determined from the NR prognostic fields using some post-processing algorithm presently unknown to us. Presumably it is an extrapolation downward from the lowest model levels. The extrapolation likely depends on the near-surface thermal structure also.

Real reports provided for some observations refer to 20 m rather than 10 m winds. In version P1, however, no vertical interpolation is performed for surface wind reports, and thus the simulated observations may have a low-speed bias for 20 m observations. Also, the extrapolation used to produce the 10 m winds in the nature run may be very different than in the GSI. This too can create biases or unrealistically large errors. The crude treatment in version P1 will be corrected in later versions.

3.9 Thermodynamic Verses Virtual Temperatures

The version of GSI used at the GMAO expects that, under particular conditions, the temperature observations within the BUFR data files will actually be corresponding values of virtual temperature Tv. Those conditions are: (1) the report contains a valid moisture observation at the same location, as required to transform between Tv and T; (2) the observation is at a level p>300 hPa. The validity of the observations is explicitly expressed by its associated quality mark in the BUFR file. For software that is not expecting Tv in place of T under these conditions, the writing algorithm for this data must be changed in subroutine read_write_obs_tq in module m_bufr_rw.

4. Adding Observation Plus Representativeness Errors

There is separate software for adding random errors to the observations to account for sums of instrument plus representativness errors This software takes the simulated observations in BUFR format and creates a corresponding file of error-added observations.

Currently, values of errors to be added are determined randomly from a probability density function (pdf). In version P1, that pdf is Gauusian. The mean is specified as zero, so no biases are added. The standard deviations are specified as tunable fractions of the corresponding error statistics used by GSI. The added errors are constructed to be uncorrelated, except for conventional observations that are provided on multiple levels for a single report, such as is the case for rawindsondes.

The standard deviations for instrument plus representativeness error used by GSI are provided on 2 files. One is for satellite radiances, which provides distinct values for each channel of each instrument on each satellite. The table provided for the P1 observations only includes values for those sets of data subtypes used, but for all channels. The file for conventional observations provides tables of values for prescribed pressure levels for each observation type. Values at observation pressure levels are linearly interpolated in pressure using the table values.

For the observations whose errors are assumed correlated in the vertical, the assumed correlation function is akin to the error function for vertical distance z. Specifically, the correlation is described in Fig 4.1.

Error Correlation Function

Denote p_1 and p_2 as pressure levels corresponding to 2 observations. d is a user-specified value that designates the value of the ratio p_1/p_2 or p_2/p_1 for which the correlation drops to a value of 0.1. Then the correlation c of errors is specified as

$$c = \exp\left[\ln 0.1 \frac{\left[\ln\left(p_1/p_2\right)\right]^2}{\left[\ln d\right]^2}\right]$$

Figure 4.1: The tunable function that describes the vertical correlation of instrument plus representativness errors.

The value for d is intended to be user set. Currently separate values are expected for wind, temperature, and relative humidity (rh) fields. Since specific humidity q varies so

greatly in the vertical, the correlations for moisture are prescribed in terms of corresponding rh by first converting q to rh, then adding vertically correlated errors in rh, and finally converting back to q. The conversions assume that values of temperature are available corresponding to each q so that values of saturation specific humidity can be determined. This determination use a functional form for saturation vapor pressure based on liquid water.

Vertically correlated errors are added by separately considering the observations for each field type as a vector (***e***), with its elements corresponding to the pressure levels at which the observations are defined. The error covariance matrix for each such vector is determined first. Then, a routine from a standard mathematics library is used to compute the positive semi-definite eigenvalues (*r*) and orthonormal eigenvectors (***v***) of each matrix. The error structures defined by different eigenvectors are uncorrelated, and the corresponding eigenvalues are the portions of total variance expressed by each such structure. Thus, appropriately correlated errors can then be produced by summing independent random contributions by each eigenvector expressed as x***v*** where *x* is a random number drawn here from a Gaussian distribution with mean 0 and variance r. The covariance of this randomly constructed sum is exactly that of the original covariance matrix. Users outside of the GMAO may have to replace the subroutine used to compute eigenvalues and eigenvectors with another available to them.

Although it is not done here, biases can be easily added. The question is then, however, what should those biases be. What may make sense for an OSSE is not clear, since there are already likely biases between the nature run and assimilating models that may be very different than those between either and the real atmosphere. Without much better estimates of the latter, it is difficult to judge the realism of the former. For this reason, any study of biases within an OSSE must be performed very carefully with only appropriate questions. Those questions could then guide the introduction of suitable additional observation error biases.

5. Validation

The software used to generate the P1 data was first tested using fake data. Interpolations were compared with hand calculated values for a broad set of locations, including values near the poles, equator, and prime meridian. Data was written and then read to make sure that all the required information was available and accurate.

The remaining testing is performed in two stages. The first is to further debug the software that simulates observations and to provide starting estimates for additional error variances that must be simulated. The second is for tuning the simulation parameters in order to produce a validated benchmark OSSE.

5.1 First Testing Procedure

The first test of the P1 data was intended to check that the observation data files could indeed be read by the GSI and that the distributions of observations appeared correct. At the GMAO currently, the software designed to examine observation data sets is designed to read binary files produced by the GSI, rather than the original BUFR files, so it is after ingestion by GSI that we can most readily examine observations graphically. By using GSI, we can also distinguish between the entire set of observations and those actually accepted by its quality control procedures.

This first test is performed by using as a background the nature run fields interpolated to the GSI grid (in our case, the GMAO version of that grid). No satellite bias correction is performed. The initial experiment ingests all the simulated observations but with no added random errors and all radiation observations are computed as cloud free. It is only necessary to perform this test for a single assimilation time.

Both the OSSE observation simulation software and the GSI produce simulated observations by applying forward models to the gridded filed input to them. Although the forward models are not identical, both employ spatial and temporal linear interpolation and versions of the CRTM. Although the gridded fields are not identical (one uses the original nature run fields directly and the other uses those fields first interpolated to a lower resolution grid), they are similar enough that the O-F=y-H(x_b) differences should be generally smaller in magnitude than for the equivalent calculation with real observations since in the simulation, instrument error is absent and both background and representativeness errors are minimized.

This test was very informative and quick since it was unnecessary to simulate weeks of data. Several minor software bugs were discovered in the observation simulation code and in the software used to create GMAO background data sets from the nature run. This test is very sensitive and quantitative, not simply relying on how some graphical representation of the data "looks."

Differences in forward models and resolutions of field data between the OSSE observation simulation software and the GSI are equivalent to errors of representativness. In order to obtain a valid OSSE, the variances of simulated representativness and instrument errors must be close to those in reality. To expedite the tuning of the software that simulates such errors for the OSSE, it helps to know what the variances are of the representativeness errors already implicit in the experiment. The required statistics are provided by the previously describe experiments: the variance of O-F is correctly interpreted as the already implicitly added variance of representativeness error. The difference between this implicitly produced variance and the value (R) assumed within GSI can then be used to define an initial guess for the fraction of R to be added by the error simulation software.

The above procedure is not valid for cloud-free radiance observations (brightness temperatures). For real IR observations, one important source of representativeness error is due to the mistreatment of a cloud affected observation as though it is cloud free. This error occurs for optically thin clouds that do not create so cold brightness temperatures that they are easily distinguished as cloud-contaminated. Some cloud affected simulated radiances will have this error implicitly. Thus, the test must be repeated with the cloud-affected radiances to derive a first estimate of the fraction of variance to be added.

Once the fractions of error variances to be added are estimated, the P1 observations that have been created without explicit random error are passed through the software that adds such error. This produces new BUFR data sets; i.e., the original data sets without such error are preserved

Of course, O-F variances for a single time (i.e., 6-hour period) may not be representative of values over the course of a month. But the values as determined above are intended only to provide starting estimates for the subsequent iterative validation procedure. That procedure is described next.

5.2 Second Testing Procedure

The second validation procedure is to determine the tuning parameters required to produce specified corresponding effects of real observations measured within a data assimilation framework. The usual metrics, where they have been employed at all, are forecast error metrics produced with and without a particular data type; i.e., conducting equivalent observation system experiments (OSEs) in the OSSE and real assimilation contexts. The forecast metrics are typically scores such as anomaly correlation coefficients or root mean squared errors. Another set of metrics, less employed, is to compare variances of analysis increments or differences in analysis with and without particular instrument types.

The production of OSEs is very expensive. Each requires a minimum of 6 weeks of simulation to allow for spin-up of the experiment and a sufficient period over which to

sample results. Since a single observation type produces only small changes in forecast metric, even 6 weeks is likely too short. The expense of this procedure is worse for this OSSE tuning exercise, because early experiments are likely to reveal problems, requiring re-tuning of the simulated observations and repeat of the tests.

In order to refrain from expensively producing many OSEs, we will instead use the adjoint-estimated forecast metric suggested by Langland and Baker (Tellus, 2004, page 189-201) and further described by Errico (Tellus 2007, page 273-276), Gelaro et al. (Meteorologische Zeitschrift 2007, page 685-692), and Tremelot (Meteorologische Zeitschrift 2007, page 693-694). Essentially, this produces estimates of forecast skill improvement due to arbitrary subsets of observations at the cost of approximately two executions of the data assimilation system over the required period (even a month appears sufficient for these studies). One can aggregate the observations not only by type but also by channel and elevation. It is thus equivalent to hundreds of OSEs.

Naturally, there is a trade-off due to reducing the number of assimilation experiments required. The principal trade-off is that only a single quadratic metric of forecast skill is being compared in an adjoint-based experiment. This metric is typically a mean squared error of the fields expressed as an "energy" norm, where the averaging is typically performed over a large volume of the atmosphere (e.g., the troposphere over the globe or northern hemisphere; see Errico (Q.J.R.M.S. 2000, page 1581-1599) for an explanation of the derivation and interpretation of this norm). If other metrics or averaging regions are to also be considered, additional adjoint-based experiments must be conducted.

The adjoint-based procedure is not identical to an OSE evaluation. They are measuring different things in different ways, so it should not be surprising if they produce different results with different conclusions. Adjoint-based and OSE results have been compared (Gelaro and Zhu 2009, submitted to Tellus), however, and have been shown to yield similar conclusions for most observation types. This comparison is invaluable for the present OSSE, because not only does it aid interpretation of adjoint results, but it also provides both adjoint and OSE results for the real data cases for July 2005 and January 2006 corresponding to our nature run periods. We therefore do not need to reproduce many of the real-data experiments with which to compare our OSSE baseline results.

An example of the observation impacts measured by Gelaro and Zhu is presented in Fig. 5.1. The score is the reduction of the global, mean squared, 1-day forecast error due to assimilation of the indicated observation types averaged over July 2005 in the GMAO analysis and forecast system. Specifically, the forecast metric is the "energy" norm (Errico 2000). It indicates, for example, that AMSU-A has the largest mean impact with an error reduction of 27 J/kg followed by rawindsonde observations with a reduction of 26 J/Kg. We will attempt to produce similar values for all these instrument types in the OSSE context.

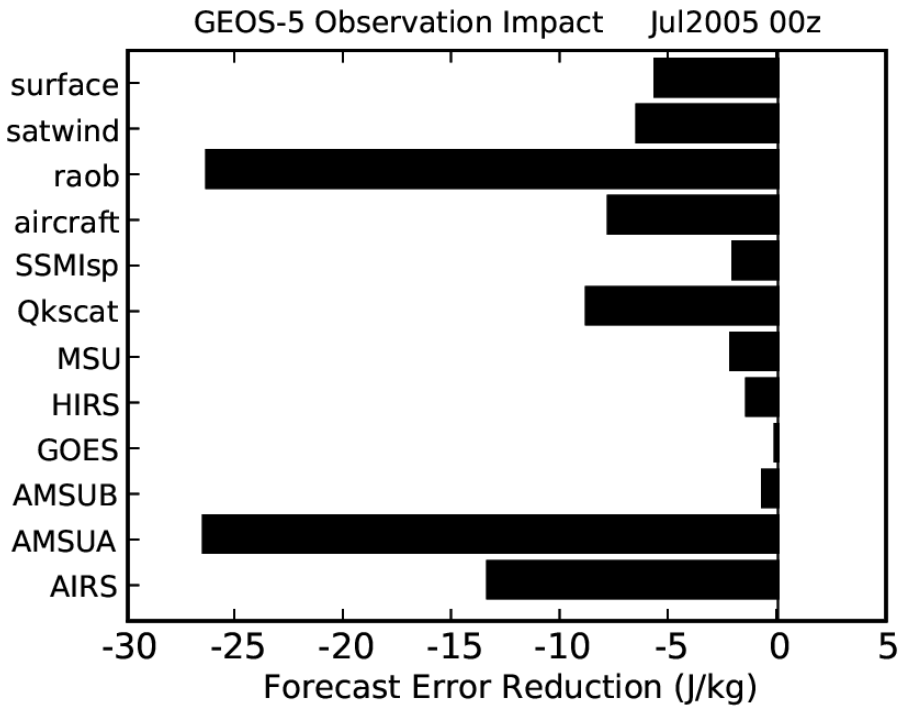

Figure 5.1: Estimates of mean reductions of 1-day forecast error in the GMAO GEOS-5 DAS, measured in terms of the energy norm (units J/kg) for indicated sets of observations (from R. Gelaro and Y. Zhu, Tellus 2009).

Our goal will be to tune the fractions of observation error standard deviations for the software that adds simulated random instrument plus representativeness errors and the parameters in the probability functions and effective sigma values for radiation-affecting clouds so that the numbers of observations accepted by the GSI quality control procedures for each type of instrument and radiative channel are similar to corresponding real acceptance rates and to match observation impacts such as shown in Fig. 5.1 rather closely. Based on resolution comparisons by Ricardo Todling, these tuning experiments can be performed at resolutions on a 2 degree latitude by 2.5 degree longitude grid. We do not need to match values exactly. Even getting within +/- 20% of the real values will be a successful validation.

It may happen that it is not possible for us to tune the presently-designed variables in order to achieve our goal. There could still be software bugs in the simulation software or those parameters may not allow us enough freedom to compensate for shortcomings in the nature run; e.g., over or under active clouds or dynamical states or the lack of consideration of biases. We just have to proceed with the tuning procedure at this point and learn what is possible. If we get stuck, radical modifications to our approach may be required. So far, however, we remain hopeful.

6. Software Design

The software is divided into three distinct functions, each with its own main program. These are software for: (1) simulating conventional (i.e., non-radiance) observations, (2) simulating satellite-observed radiances, and (3) simulating random added instrument plus representativeness errors. These software have many common sub-components that are all placed in modules. Specific purposes of the programs are controlled via an input argument list. Other user-specified values are provided through resources files to be read. There should be no need for the user to make any changes or selections within the FORTRAN program or modules themselves.

6.1 List of Modules

Subroutines called by more than one program have been placed in modules. Each is listed and described individually below. Information that is only required by the subroutines within any single module and that does not need to be passed back to the calling program is kept within the module. Some such information, such as required for dimensioning arrays found only in the module, is copied from the calling program to the module in setup routines.

The modules are:

m_bufr. This module contains all subroutines for reading and writing BUFR data compatible with the GSI for all the simulated observations. It also includes a function (check_type) that contains lists of observation subtypes to include.

m_clouds. This module contains all subroutines pertaining to the determination of whether clouds are present affecting radiance transmission at an observation location.

m_interface_crtm. This module is an interface between the CRTM and the main program for simulating radiances. It includes determination of variables that are specifically required by the CRTM but not by the main program.

m_interp_nr. This module contains all the routines for horizontal, vertical, and temporal interpolation for either surface information, single level data, multiple level data, (e.g., rawindsonds) or profiles (e.g., as required to produce satellite radiances). It also contains software for reading required nature run fields. See below for further information regarding reading and storage of the nature run fields.

m_kinds. This module specifies variables for the various kinds of real variables used by the software. See below for further information regarding the motivation for using various kinds.

m_obs_pert. This module contains subroutines for adding random errors to each observation report. Note that it uses a library routine to compute eigenvalues and eigenvectors of a covariance matrix.

m_rdata_boxes. This module contains all subroutines concerned with radiance data thinning.

m_relhum. This module contains all subroutines for transforming between relative and specific humidity.

6.2 Kinds of Real Variables

The software allows for three kinds of real variables. One kind primarily concerns storage of the nature run fields, another observation values, and a third all other variables. The intention has been to allow variables that do not need high precision, such as the nature run fields that are stored in data files as packed GRIB data, to be stored as 4-byte values rather than 8-byte ones. On the other hand, some other variables must be treated as 8-byte ones, notably some arguments in calls to the BUFR library. Since the nature run 3-d fields contain so many values, storing them as 4-byte ones permits use of a single processor in version P1; otherwise, in general, multiple processors would be required to hold the data arrays in memory.

6.3 Storage of field arrays

Observational data for the GSI are stored on files containing reports over 6-hour periods centered on 0Z, 6Z, 12Z, and 18Z. The nature run fields are provided every 3 or 1 hours for the T511 and T799 data sets, respectively. Thus, observations within any 6-hour period being considered are interpolated from two corresponding times in the nature run. The interpolation software reads all the times (either 3 or 7) relevant for the 6-hour period into memory, so that all are available as the software loops through the observation reports. All two-dimensional fields are stored in a single array. Likewise, all 3-d fields are stored in a single array.

The size of the 3-d field array can become quite large if many such fields or times are required or if the T799 fields are used. Some compilers or machines may not allow such large arrays. Another version of the module m_interp_nr is available that breaks this single array into three, one for each time considered for the T511 data set. This can be used, but is more limited with regard to the numbers of 3-d fields, times or resolutions that can be considered. For that reason, it is not recommended.

The software is designed to only place in memory those specific 2-D or 3-D fields that are required for any specified purpose of the interpolation software. These fields are identified in an array (field_names) that contains the names of the fields required. The

fields themselves are placed in arrays with generic names such as fields_2d. The software identifies what is stored in each part of an array according to the order of names in the field_names array. When it needs to find a particular field such as u or ps, it searches through the list of names until it locates that name. This action defines an index that is then used to indicate particular portions of the fields arrays. If a required name is not found, execution stops with an error message, unless the software is instructed that the problem is not a fatal one and execution should continue.

The nature run fields are stored on the reduced Gaussian grid. This reduces memory requirements but grid indexes for particular latitudes and longitudes must then be determined by an algorithm. This uses an array (nlonsP) of pre-computed index values for the index for the last longitude in the adjacent latitude to the south. Longitudes are stored east to west starting at the prime meridian and latitudes are ordered from south to north.

The fields on the reduced Gaussian grid are actually augmented by including values at the poles so that no interpolations are required to pass over the poles.. Although these additional field values are at the pole, they are specified for the same number of longitudes as for the Gaussian latitudes adjacent to the poles. For the ECMWF reduced grid, this number is 8. For all fields but the wind, all the values at each pole are specified as the mean of the values for the same field and vertical level as the Gaussian latitude adjacent to the pole. For the wind field it is specified as the average of the zonal wave number 1 Fourier coefficients for the two wind components, accounting for a pi/2-phase shift of v with respect to u. The approximation here is that there is no meridianal gradient of the zonal wave number 1 component of the wind as the pole is approached from the adjacent latitude. For further details, consult the software.

The 3-D fields are also augmented by including near-surface values in addition to those on the above-surface atmospheric levels provided. Thus, for the ECMWF data, they are stored for 92 levels, rather than for 91. For the wind field, these near-surface values are the 10m winds provided in the nature run data set. For temperature, they are the values of T at 2 meters. For specific humidity, they are computed from this T and the 2m dew-point temperature.

The array of 3-d fields for the T511 dataset is specified by 32,067,888 values for each time and field. The storage required for three times using 4 bytes per value is approximately 385 MB per field. In version P1, the total memory required when simulating conventional data is 0.8 GB because only two 3-d fields are required (since simulations of wind (u,v) and mass (T,q) are performed separately. For radiance data, 3-d fields of T, q and ozone are stored simultaneously, and 1.4 GB of memory is required.

6.4 Interpolation Search Algorithms

When an interpolation to the latitude of an observation is to be performed, it is not trivial to determine which two Gaussian latitudes sandwich the desired latitude, since they are

not equally spaced. The software exploits the fact that they are almost equally spaced however, with the latitudes closest to the poles offset from the pole by approximately ½ of the spacing between other consecutive latitudes. Using this approximation, the software computes a range of indexes for a set of Gaussian latitudes to inspect to determine which two are closest to the desired observation latitude. In this way, it need not look through all the latitudes until the desired one is found. This search algorithm was tested for many observation latitudes and appears to function as intended.

The vertical spacing of pressure levels for the nature run grid is also not uniform. In fact, within the troposphere the spacing varies with surface pressure since the vertical coordinate is a hybrid (mixed sigma and pressure) one. When interpolation to a specific pressure is to be performed, the pressure levels sandwiching it are identified by searching through the pressures defined for all the levels. In order to accelerate this search process, however, the search algorithm uses an iterative strategy of dividing ranges of possible vertical level indexes by two and identifying which half the desired level is in. In this way, only log_2 (K) + 2 iterations are required to search through K values of level pressures.

6.5 Nature Run Data Files

The current software does not read directly from the ECMWF GRIB files. Instead it is reading from binary files with a special format. Each file of 3-D fields contains a single field (e.g., u) at a single time, on the reduced Gaussian grid. The interpolation software thereby only reads the files containing the fields it requires, as specified in the field_names array. A user who does not want to first create these binary files from the GRIB data needs to replace the nature run reading routines in the module m_interp_nr.

6.6 Interpolation of Humidity

The software is designed to either interpolate humidity vertically in terms of specific humidity or relative humidity. The latter is usually preferable because it changes less rapidly in the vertical. Since the nature run data extend into and above the upper stratosphere, however, the transformations between specific and relative humidity that are used by the interpolation software fail (e.g., yielding negative humidity values). An option exists for only making such transformations below some level in the atmosphere, but the subroutine in which vertical interpolations are performed is not made aware of this discontinuity in the meaning of the humidity field, and thus will yield an erroneous vertical gradient between the transition levels. In order to avoid this confusion, the option of using relative humidity is therefore not used in version P1 (the logical variable l2qrh is set to .false.). Transformations to relative humidity are however made when considering vertical correlations of simulated added instrument plus representativeness errors for conventional observations, since it is assumed that no insitu moisture observations are available above 10 hPa. No check of that assumption is made, however, so the user should be aware of this limitation in the P1 software.

6.7 Changing resolution

In version P1 of this software, some variables have been preset to those required for the T511L91 ECMWF data set on the reduced Gaussian grid. To run with a different resolution, Lat-Lon grid, or nature run output intervals, several changes must be made. First, a different file ossegrid.txt, as described in section 7.3, must be prepared. Also, the following variables must be reset in the subroutine setup_m_interp in the module m_interp_nr: nfdim, nlevs, nlats, and ntimes.

Lastly, values of the array time_files must be set to time, in hours, relative to the central time of the period for which observations are being simulated. At NCEP and the GMAO, this period is 6 hours and the central times are the synoptic times 0, 6, 12, and 18 UTC, corresponding to the organization of the observational data files. For the T511 ECMWF nature run that has fields provided at 3 hour intervals, the array time_files has the three values –3., 0., and 3.

7. Resource Files

There are 2 resource files that are to be user specified. These all involve specification of variables used for tuning the observation simulation. We recommend using, or at least starting with, the resource file values provided with the software.

7.1 The File cloud.rc

One resource file is cloud.rc. It specifies parameters used by the program that creates simulated radiance observations from the nature run. These parameters are independently specified for AIRS, HIRS (values for both HIRS2 and HIRS3 are treated as identical), and AMSU (values for AMSU-A, AMSI-B, and AMSU-A on AQUA treated as identical). A sample cloud.rc file appears in Fig. 7.1.

```
AIRS
   ncloud    3  irandom 1111 box_size   60
  c_table
 high cld  hcld  0.10  0.40  0.70  0.30
  med cld  mcld  0.10  0.40  0.70  0.60
  low cld  lcld  0.10  0.40  0.70  0.90
HIRS
   ncloud    3  irandom 1221 box_size   90
  c_table
 high cld  hcld  0.10  0.40  0.70  0.30
  med cld  mcld  0.10  0.40  0.70  0.60
  low cld  lcld  0.10  0.40  0.70  0.90
AMSU
   ncloud    4  irandom 1331 box_size   90
  c_table
 land msk  almk  0.10  0.10  0.10  0.70
  ice msk  ismk  0.10  0.10  0.10  0.70
 c.precip  conp .0002 .0002 .0002  0.50
 s.precip  rain .0002 .0002 .0002  0.70
```

Figure 7.1: A sample cloud.rc file

The integer following "ncloud" refers to the number of distinguishing fields that are to be considered when determining a probability function that characterizes whether an observation is affected by clouds in the case of AIRS and HIRS or by surface characteristics or precipitation in the case of AMSU. For the IR radiances observed by HIRS and AIRS, the distinguishing fields are the cloud fractions for three height ranges of clouds. For AMSU, these fields are the land and ice fractions and the precipitation accumulations at the surface over the time span between ECMWF data output times.

The integer following "irandom" is used to help set the seed for the random number generator used for the cloud determination algorithm. The seed is given by the sum of this number and an integer representing the central date and time of the dataset being

produced (YYYYMMDDHH). In this way, different instruments and dates use different sequences of random numbers.

The integer following "box_size" denotes the approximate width and length, in units of km, for a "thinning box" on the globe. The smaller the size of the box, the less data will be thinned, the more calls to the CRTM will be required, the more observations will be provided to the GSI for it to then apply to its own thinning algorithm, and the more cloud-contaminated observations there will be. The latter results because the thinning algorithm favors observations that are less cloud affected within each box, but if there are fewer observations to compare within a box, the more likely a cloudy one will be retained. Currently, the GSI has thinning boxes of approximately 180 km, so using 90 km here means approximately 4 observations will be provided to the GSI from which it will choose one. A value of 60 km here implies approximately 9 will be provided.

In the table for high, medium and low cloud parameters, the 4 values in each row are the a, b, c, and sigma that define the cloud probability function and effective cloud top, if a cloud is present. Adjusting these values will change the numbers of observation channels accepted as cloud free by the GSI quality control algorithm. For example, increasing the values of sigma in each category will increase the numbers of channels accepted, since then cloud tops are lower in the atmosphere and there will tend to be more channels that peak enough higher as to be relatively unaffected by those clouds. Note that sigma for high clouds should be between 0.1 and 0.45, for medium clouds between 0.45 and 0.8, and for low clouds between 0.8 and 1.0, in accordance with the definitions of the cloud fractions in the nature run data set. For the a, b, and c parameters, the user should examine the description of the cloud probability function as well as histograms of cloud fraction values in the nature run and then carefully consider what values may be appropriate and useful.

For AMSU, cloud effects are ignored but effects of precipitation and the uncertainty in surface emissivity are considered. The land mask and ice mask are examined and if the observation location is not an ocean one, then the simulation software is instructed to contaminate the observation by misplacing the surface at sigma=0.7 rather than at 1. If the precipitation rate for convective or stratiform precipitation is sufficiently, then the microwave signals are treated as contaminated. Convective precipitation is considered to occur at a lower pressure than stratiform.

This resource file is read within the module m_clouds. If it is changed, care must be taken to do so in accordance with the proper FORTRAN format. If a user has any doubts, subroutine set_cloud should be consulted in that module. The orders of presentations of the fields to be examined is important, since if an effect is determined for the first examined, the remainder are not even considered. So, for example, if a high cloud is found to be present, there is no need to consider clouds beneath it.

7.2 The File error.rc

The resource file error.rc specifies parameters used by the software that adds simulated random instrument plus representativeness error to the observations. It is read within the subroutine read_error_rc that is called by the program add_error. An example appears in Fig. 7.2

The file is divided into separate sections for some distinct observation types. Each such section is separated by a blank line followed by a line of underscores. Each section begins with a data type name. The first section begins with a line describing some formats to follow. The remaining lines in that section are for some variables shared by all observations.

Two of the specified variables are used to help create the seed used to initiate the sequence of random numbers to be used to generate random errors. The seed is specified as the sum of three integers. These are:

idatetime: a 10 digit integer containing the date and hour defining the central time of the period of observations in the data set. The integer's format is YYYYMMDDHH. This value is passed to the program as an argument from the UNIX script invoking the program. The use of this value is intended to provide different seeds for data sets valid at different times.

random_case: a 1-5 digit integer that allows the user to define different seeds for different cases; e.g., if an ensemble of observation data sets are to created. Each data set would then be created with a different sequence of random numbers. For non-ensemble applications, this value can remain unchanged.

random_type: a 1-5 digit integer used to specify a different seed for each observation type.

The value of pert_fac specifies a tunable parameter that defines what fraction of the standard deviation of instrument plus representativness error (square root of R) should be used to define the standard deviation of the Gaussian probability function from which random errors are drawn. In version P1, this fraction is specified identically for all observations of the indicated types. So, for example, wind observations provided by both rawindsondes and cloud tracks will use the same fraction. The actual error standard deviations for these two diverse types will differ, however, because R read from the GSI tables differs. The limitation is that, in this version, the fraction cannot be tuned differently for each.

The correlation distances specified in the resource file refer to the parameter d used to define vertical correlations for multiple-layer conventional observations such as rawindsondes. This has been discussed in section 4.

```
var_name_____a16  nnnnn
number of lines      52   ! number of lines in this file
random_case         333   ! This number is used to modify the random seed

_____
WIND_
pert_fac           0.70
random_type        1111
corr_distance u     1.2
corr_distance v     1.2
file_err_table    conv_err_table.txt

_____
MASS_
pert_fac           0.70
random_type        2222
corr_distance t     1.2
corr_distance q    1.09
file_err_table    conv_err_table.txt

_____
AIRS_
pert_fac           0.70
random_type        1221
corr_distance 0      0.
corr_distance 0      0.
file_err_table    /sat_err_table.txt

_____
HIRS2
pert_fac           0.70
random_type        1223
corr_distance 0      0.
corr_distance 0      0.
file_err_table    /sat_err_table.txt

_____
HIRS3
pert_fac           0.70
random_type        1225
corr_distance 0      0.
corr_distance 0      0.
file_err_table    /sat_err_table.txt

_____
AMSUA
pert_fac           0.70
random_type        1227
corr_distance 0      0.
corr_distance 0      0.
file_err_table    /sat_err_table.txt

_____
AMSUB
pert_fac           0.70
random_type        1229
corr_distance 0      0.
corr_distance 0      0.
file_err_table    /sat_err_table.txt
```

Figure 7.2: A sample resource file error.rc

The first line in the resource file is simply a reminder about the format of the following lines. The variable name is restricted to 16 characters and the follwing values to 5 digits.

7.3 The File ossegrid.txt

The file ossegrid.txt contains some information about the nature run grid. In particular it contains the arrays of ak and bk that define the hybrid vertical coordinate on interfaces of the 3-d grid, such that the pressure at each interface location is p=ak+bk*ps, with ps being the local surface pressure. Also, the number of longitudinal points at each latitude of the reduced (non-cartesian) horizontal grid are read as are the Gaussian latitudes themselves.

From values read from this file, a table of beginning indices for data at each latitude is determined (see section 6.3). Also, note that the values of ak and bk read are at once replaced by analogous values valid at data levels, computed by simply averaging adjacent interface values.

8. Instructions for Use

The current software has 3 executables. The use of each is described separately below. All three use the NCEP BUFR software library for reading and writing observational data sets in BUFR format. Additionally, the software to simulate radiances uses a JCSDA CRTM library. The executable sim_error.x also requires a routine to compute eigenvalues and eigenvectors. In version P1, all of the executables are designed to be executed on a single processor that must have at least 1.5 GB of memory.

In this section, the function and arguments of the executables are described. Users should also acquaint themselves with the proper interpretation of printed output from these programs to confirm that executions are successful. That printout is described in section 9.

8.1 The Executable sim_obs_conv.x

This executable produces either wind (u, v) or mass (T, q, ps) observations. Although labeled here as "conventional" observations, they include, for example, cloud-track winds and surface winds "observed" from satellite. Specifically, they include all observations provided as the mentioned field variables but not those provided as radiance measures expressed as brightness temperatures. On the main-frame computes at NASA, creating either all the mass or all the wind observations for a typical 6-hour simulation period requires about 30 seconds of single-processor CPU time.

Expected arguments are d_type, input_file, and output_file, in that order. If exactly three arguments are not provided, execution will stop with an error message indicating what arguments are expected.

The first argument has the value WIND_ or MASS_, indicating which class of observations are to be simulated. In version P1, these are simulated separately to limit the processor memory required. If neither of these acceptable values is presented for this argument, an error message will be printed and execution will stop.

The second argument is the name of the input file that provides the observation locations and a template for the file of simulated observations to be created. Generally, this should be a GSI .prepbufr file, containing a pre-processed data set of conventional observations for the same date and time period corresponding to those for the observations to be simulated. This file is in BUFR format and contains the required BUFR table describing its content.

The last argument is the name of the output file that will contain the simulated observations to be produced. It will be in BUFR format, in a form to be read by the GSI. As described in section 3, it is only guaranteed to contain that information actually required by GSI; i.e., ancillary information typically found in such BUFR data but not

actually read by GSI may be absent. Two distinct files of conventional observations will be produced, one for all conventional wind information and one for all mass information. The GSI must be notified about this distinction, because otherwise it will expect all such information to be provided in a single .prepbufr file.

Note that during execution, there is no check of the consistency between the date of the observations used to define locations for the simulation and the date of the nature run fields. This allows use of observations taken at different times as templates for the observation locations with no modifications to the software. Users must therefore check the printout from executions to insure that they are processing the data at the times they expect.

8.2 The Executable sim_obs_rad.x

This produces files of simulated radiances for all instrument types presently considered. Expected arguments are d_type, c_datetime, rc_file, input_file, and output_file, in that order. If exactly 5 arguments are not provided, execution will stop with an error message indicating what arguments are expected. On the main-frame computes at NASA, creating either all the mass or all the wind observations for a typical 6-hour simulation period requires about 30 minutes of single-processor CPU time. Most of the computation is performed within the CRTM

The first argument has one of the values HIRS2, HIRS3, AMSUA, AMSUB, or AIRS_, prescribing what group of radiances is to be simulated. If none of these acceptable values is presented for this argument, an error message will be printed and execution will stop These specific groups are in one-to-one correspondence with the files containing these same groups as used at NASA. For the first four groups, observations with that instrument on any satellite used are simulated in that execution. Specification of AIRS_ signifies that both the AIRS and AMSU-A observations on the AQUA satellite are to be simulated, since for the GSI, both these observation data sets are included in the same report.

The second argument is the name of resource file cloud.rc that controls data thinning and cloud, precipitation, and surface effects on radiation (see section 7a).

The third argument is the date and time presented as a character string of integers describing the date and time as YYYYMMDDHH. It is used to specify the seed for the random numbers employed to determine whether the presence of clouds or precipitation is affecting radiation transmission through the atmosphere. No check is performed to ensure that this time and the times on the nature run or observation data files are consistent (for the reason described in section 8.1).

The fourth argument is the name of the input file that provides the observation locations and a template for the file of simulated observations to be created. This file must be in a

BUFR format expected by GSI. Except for data type AIRS_, it is expected that the required BUFR table describing its content is appended to the file.

The last argument is the name of the output file that will contain the simulated observations to be produced. It will be in BUFR format, in a form to be read by the GSI. As described in section 3, it is only guaranteed to contain that information actually required by GSI; i.e., ancillary information typically found in such BUFR data but not actually read by GSI may be absent. These output files are in one-to-one correspondence with the input files just described.

8.3 The Executable add_error.x

This produces files of simulated observations with random errors added to simulate instrument plus representativeness errors. Expected arguments are d_type, c_datetime, rc_file, input_file, and output_file, in that order. If exactly 5 arguments are not provided, execution will stop with an error message indicating what arguments are expected. On the main-frame computes at NASA, creating a file for all observations within a 6-hour period for any data type requires less than 1 minute of single-processor CPU time. Very little memory is required.

The first argument has one of the values WIND_, MASS_, HIRS2, HIRS3, AMSUA, AMSUB, or AIRS_. If none of these acceptable values is presented for this argument, an error message will be printed and execution will stop. These specific groups are in one-to-one correspondence with the files containing these same groups of simulated observations produced by the software described earlier in this section.

The second argument is the name of resource file error.rc that controls the seed for the random number generator and the fraction of variance used to create random. This file is further described in section 7b.

The third argument is the date and time presented as a character string of integers 1-9 describing the date and time as YYYYMMDDHH. It is used to help specify the seed for the random numbers employed that is used to create random errors.

The fourth argument is the name of the input file that provides the simulated observations prior to the simulated errors being added. This file is in the BUFR format expected by GSI.

The last argument is the name of the output file that will contain the simulated observations with their errors added. It will be in BUFR format, in a form to be read by the GSI. As described in section 3, it is only guaranteed to contain that information actually required by GSI; i.e., ancillary information typically found in such BUFR data but not actually read by GSI may be absent.

9. Run-Time Messages

There are 4 kinds of output printed to standard output by the executables described in the previous section. Most important are tables printed at the end of each execution that summarize the numbers of observations simulated and some of their characteristics. Another is information printed prior to the tables that describes how processing is proceeding. A third are error messages that only appear when describing why an execution is prematurely terminating. A last set are additional information that can be requested when checking the algorithms and computation in some subroutines. All four of these types of messages are described in separate subsections below.

9.1 Summary Tables

An important portion of the printed output produced by the simulation software is the summary table. These present counts of either "observations" or "reports." In this context, a single observation refers to a single value among possibly many values provided by an observing instrument associated with some geographical location. The collection of those many values constitutes a single report.

How observations are specifically grouped into reports is defined by the BUFR file formats containing the data. For example, a single report of a satellite instrument observing radiances includes values of brightness temperature for the entire set of channels at a single observing location provided to the data assimilation system. A cloud track wind report normally includes 2 observation values, one for each wind component at a single location. A single rawindsonde report contains values of T, q, ps, u, and v for all mandatory and significant pressure levels provided from one balloon ascent. In this context, the number of observations is the total number of independent T, q, ps, u, and v values in the report.

9.1.1 Table for conventional observations

A sample table printed at the end of execution of the software for producing conventional observations of T, q, or ps appears in Fig. 9.1. The number of reports read from the input file of corresponding real observations that are of the data types being considered for production appears as "observation reports read" for the data subtypes listed in the function check_types appearing at the end of the module m_bufr_rw. This is followed by the number of reports not considered because the reports have no data or are not of the type requested (e.g., rawindsonde reports containing wind information rather than mass information, as requested). The difference between this and the total number read is the number having some data of the requested type. This latter number is also presented as a fraction of the total number read for all subtypes requested. For the NCEP .prepbufr file

excluding precipitation reports as in this example, slightly less than one half of the reports are for MASS_ with the remaining fraction for type WIND_.

The number of observation values for independent fields and pressure levels summed for all reports having data to be considered is printed next. Some of these observations may be unsuitable for simulation because, for example, their times, latitudes, longitudes, or pressures may be out of range. The number of such unsuitable observation values is subtracted from the total, and the result is expressed as a fraction of the total observations values considered. This fraction will generally be close to 1.

If particular problems regarding some reports are detected while processing, an additional table of detected errors is printed. The specific kinds of tests performed on the reports are indicated along with their corresponding error counts. This includes numbers of reports whose observation times are outside the period being considered, or that have longitudes outside the range -180 through 360 degrees or latitudes outside the range -90 through +90. Due to preprocessing of the data, the latter two error numbers should be 0 but sometimes a few reports are a few seconds outside the expected time range. Any reports with such detected errors are excluded from consideration.

Those error numbers are then followed by the number of observation values associated with pressure levels above the top of the nature run data set (ptop=1.5 Pa). Generally this is 0, but any such observations in a report would be excluded from consideration (replaced with the missing-value indicator). Any valid observations in such a report would still be simulated.

```
SUMMARY TABLE:
  184902 observation reports read
   91143 number of reports without data or not requested data types
   93759 number of reports having some data of requested types
 0.50707 fraction of reports read having requested data types
  189848 number of observation values considered
 0.99243 fraction of obs values simulated vs. read for requested types

  Summary of bad observations or other errors detected:
       0 observation reports found where t<tmin
       0 observation reports found where t>tmax
       0 observation reports found where longitude out of range
       0 observation reports found where latitude out of range
       0 observation values found where obs_plev < ptop
    1438 observation values where obs_plev > ps lowest level
    1438 observation values ignored for various detected problems
       0 errors detected in writing buffer records
```

Figure 9.1: An example summary table for conventional observations data type MASS_.

Similarly printed is the number of observations reported with pressures that place them below the surface of the nature run at their respective locations. This does not include observations specifically indicated in the BUFR records as surface values. In the latter

case, the pressure levels for the surface recorded for the real observations are simply replaced by those interpolated from the nature run. An example of an error, however, would be a rawindsonde observation that is indicated as above the surface of the real atmosphere but below the surface of the nature run. Such observations are replaced by missing values. The total number of independent observation values being excluded is then printed. Rejection of an entire report may result in rejection of multiple observation values.

Only one test is performed while writing the BUFR files. This is a check that the number of values actually written in each report record is the same as the number of values requested to write. Generally, this error count is 0.

9.1.2 Table for radiance observations

A sample table printed at the end of execution of the software for producing radiance observations appears in Fig. 9.2. The example is for AIRS since, for this data type, both the usual plus some additional output is produced. This occurs because the AIRS files contain observations from both the AIRS and AMSUA instruments on the AQUA satellite in a single report.

Three integer numbers are presented. The first is the total number of reports read from the input file that are of the requested subtypes. These are all the specific subtypes listed in the function check_types included in the module m_read_bufr for the user requested data type. This is followed by the number of thinning boxes in which no observations were located. For this count, thinning boxes for independently considered subtypes are considered as distinct; e.g., if three satellites hosting the instrument are considered as distinct subtypes, then the total number of boxes considered is three times the number of boxes covering the earth. The last integer printed is the number of observation reports actually simulated and therefore written. The sum of these last two numbers is the total number of distinct thinning boxes considered, since each box contains either 1 or 0 reports.

Two fractions are printed at this point. One is the fraction of thinning boxes containing an observation. This is computed as the number of simulated observation reports divided by the number of distinct thinning boxes, with the latter counting boxes for independent subtypes as distinct. For boxes whose span is greater than the spacing between observations but not greater than scanning-swath widths, this fraction should be approximately the average of the fractions of the earth's surface covered by the swaths for each observation subtype during the observation period considered.

The fraction of reports written out vs. read in is determined primarily by the size of thinning boxes specified by the user. If at least one observation falls within a box, a report will be simulated for that box, but at most one observation is simulated for any thinning box. Appropriate specification of the thinning box size is part of the simulation tuning process. It is therefore important that the simulation data thinning procedure and its tuning be understood as explained in sections 3.4 and 7.2.

Finally, elevation of the effective emitting surface, to crudely account for clouds in the case of IR measurements or precipitation, land, or ice in the case of MW measurements, as described in section 3, is summarized in another table. Getting reasonable numbers for this table requires appropriate tuning of the cloud.rc file. Unfortunately, at this time we have too little experience to suggest what reasonable values should be for any particular data type.

```
SUMMARY TABLE;
  81000 observation reports read for AIRS_
  35729 number of empty thinning boxes of all sub-types
 0.4305 fraction of non-empty boxes
  27013 number of observation reports written out
0.33349 fraction of reports written out vs. read in
  Fractions of simulated observation with surface set as:
 0.4272 have surface as actual NR surface
 0.2212 have surface set as   1.000 > sigma >=  0.800
 0.0000 have surface set as   0.800 > sigma >=  0.600
 0.1101 have surface set as   0.600 > sigma >=  0.400
 0.2415 have surface set as   0.400 > sigma >=  0.200
 0.0000 have surface set as   0.200 > sigma >=  0.000

  Summary of AMSUA simulated data on AIRS (AQUA) file
  27013 thinned observation reports considered
  27013 number of AMSU reports written out
  Fractions of simulated observation with surface set as:
 0.4480 have surface as actual NR surface
 0.0000 have surface set as   1.000 > sigma >=  0.800
 0.0000 have surface set as   0.800 > sigma >=  0.600
 0.0214 have surface set as   0.600 > sigma >=  0.400
 0.1717 have surface set as   0.400 > sigma >=  0.200
 0.3447 have surface set as   0.200 > sigma >=  0.000
```

Figure 9.2: An example summary table for radiance observations of data type AIRS_.

9.2 Other Normal Run-Time Information

It should be sufficient to peruse the summary tables printed at the end of each execution of the observation simulation software to check whether it appears successful. Prior to those tables, however, other information is printed. This provides a record of some input values specified by the user or read from files. It also assists identification of problems that may cause an unsuccessful execution, as when input files have not been appropriately specified by the user.

9.2.1 Print regarding simulation of conventional observations

The printout begins by echoing the data type specified by the user as an argument to the executable. This then determines the 2-dimensional and 3-dimensional fields required from the nature run data sets. Some information about those fields is printed:

nlevs1: One plus the number of levels on which 3-d fields are defined. This sum is 92 for the ECMWF data at L91 resolution.

nlats2: Two plus the number of latitudes on which the nature run fields are defined. The addition of 2 is for the field values at the poles that are not among the latitudes in the ECMWF data sets. This sum is 514 for the ECMWF data at T511 resolution.

nfdim: The number of grid-point values for each field at each level in the nature run data set. This value is 348564 for the ECMWF data on the reduced, linear Gaussian grid at T511 resolution, after augmentation by the additional values for the poles.

nfields2d: The number of 2-d, nature run fields required by the simulation software.

nfields3d: The number of 3-d, nature run fields required by the simulation software.

f_names: The names of the 2-d followed by 3-d fields required from the nature run.

The file ossegrid.txt is described in section 7.3. It contains information about the structure of the nature run grid. Some additional required arrays are computed from this information as indicated in the printout.

A table of saturation vapor pressures is created for computationally efficient conversions between specific humidity and relative humidity. This table is stored as an array satvp.

Next the required fields from the nature run are read as indicated. Then pole values are created by extrapolation from the nature run fields provided, as describe in section 6.3. Also, values of specific humidity at the surface are created from values of dew-point temperature at the surface provided in the nature run data set. The setup of the nature run fields is then indicated as complete.

The input and output file names will likely be generic ones specified in the script calling the executable, but linked to actual files of real observations read in and simulated observations written out. The list of observation types processed, as determined by what is actually present in the input file and what has been included in the list provided in the fiunction check_type in the module m_bufr_rw. The intention here is that for normal executions, all observations that the software can simulate will be processed, so the user generally will not need to change the list in this function except as the rest of the software is updated.

```
Begin processing for type=MASS_

 Setup_m_interp for nlevs1, nlats2,  nfdim, nfields2d, nfields3d
                        92     514  348564          2          2

  f_names= pres zsfc temp sphu

 File=ossegrid.txt opened for reading grid info on unit= 10

  Table for nlonsP filled

  Grid information set

  Table for satvp filled in module m_relhum setup

  Begin read of NR data
 File=pres_NR_01 opened for reading ps data on unit= 12
 File=tdat_NR_01 opened for reading 3D data on unit= 12
 File=qdat_NR_01 opened for reading 3D data on unit= 12
 File=surf_NR_01 opened for reading surface data on unit= 12
 NR fields read for 1 times

  [REPEAT OF ABOVE FOR NR_02]
  [REPEAT OF ABOVE FOR NR_03]

  Pole values set
  td converted to q at surface
 Setup of NR fields completed

 input file=conv.bufr opened on unit=  8

 output file=obsout4.bfr opened on unit=  9
Processing subset ADPUPA   for datetime 2005120106
Processing subset AIRCAR   for datetime 2005120106
Processing subset AIRCFT   for datetime 2005120106
Processing subset SATWND   for datetime 2005120106
Processing subset PROFLR   for datetime 2005120106
Processing subset VADWND   for datetime 2005120106
Processing subset ADPSFC   for datetime 2005120106
Processing subset SFCSHP   for datetime 2005120106
Processing subset SPSSMI   for datetime 2005120106
Processing subset GOESND   for datetime 2005120106
Processing subset QKSWND   for datetime 2005120106

  [SUMMARY TABLE PRINTED HERE]

  Grid arrays and fields deallocated

 Program completed
```

Figure. 9.3: Standard printout from execution of sim_obs_obs.x for data type MASS_. The sections in square brackets have been omitted to fit the table on a single page, but the summary table appears in Fig. 9.1.

```
Set cloud table
 input file=cloud_withcld.rc opened on unit= 16
   ncloud    3  irandom 1111 box_size   90
 c_table
 high cld  hcld  0.10  0.40  0.70  0.35
  med cld  mcld  0.10  0.40  0.70  0.65
  low cld  lcld  0.10  0.40  0.70  0.85
Seed for random number generator =   2006011511  idatetime=  2006010400
Cloud table and indexes filled for AIRS_

 Thinning boxes defined for      62742 boxes
 box_size=    90.0, nlats,dlat= 222  0.81, ntypes=  3
  Additional thinning box created for storing satellite spot info:
 n_spot2= 25, nboxes=     62742

 input file=airs_bufr_table opened on unit=  15

 input file=airsY.bufr opened on unit=  8
 Processing subset NC021250 for date 2006100100

  Numbers of profiles to be considered for each subtype:
  27013  27013  27013
  Indexes of detected subtypes:
      1      2      3
```

Figure. 9.4: Standard printout from execution of sim_obs_rad.x for data type AIRS_. The sections in square brackets have been omitted to fit the table on a single page, but the summary table appears in Fig. 9.2 and other information in Fig. 9.3.

9.2.2 Print regarding simulation of radiance observations

The information printed prior to the summary tables when simulating radiances includes that printed when conventional observations are produced (section 9.2.1), plus some additional information that is described in this section.

Information read from the cloud specification resource file (section 7.1) is echoed in the print out. This includes the name of the file read. Section 3.1 should be consulted for a description of this cloud information.

Information about the data thinning boxes (section 3.4) is printed next. This includes the number of boxes created, covering the globe, the size of the edges of each box (measured in kilometers, as requested by the user), and their arrangement (number of latitudes and spacing between latitudes, in units of degrees). For instruments other than AIRS, the variable ntypes is equal to or greater than the number of satellite platforms hosting that instrument. These must be distinguished because the spectral coefficient tables for the fast radiative transfer algorithms sometimes differ with satellite. For AIRS, ntypes=3 distinguishes the 3 instruments (AIRS, HSB, AMSUA) combined in the same reports in AIRS BUFR files. All the different instruments or satellites are kept distinct, in their own sets of thinning boxes.

The number of thinning boxes containing a report is printed for each satellite or instrument. A box will contain a report if at least one observation falls into that box for that subtype. In the case of AIRS, because reports of all instruments are combined, all three subtypes have identical numbers. For instruments on NOAA satellites, the subtypes 1-5 correspond to the platforms NOAA 14-18. Only values for non-empty sets of boxes are printed, along with the indexes for those particular subtypes.

9.3 Error Messages

At this time, very few error messages are printed. Those that are should be self explanatory, but they may require examination of the portion of code near where the print command is issued.