EXPERIENCE AND LESSONS LEARNED REGARDING CONFIGURATION AND CONTROL OF AN ADVANCED 4 DIMENSIONAL VARIATIONAL SATELLITE DATA ASSIMILATION SYSTEM

P2.5

Scott Longmore*, Andrew S. Jones, Adam Carheden, Thomas H. Vonder Haar
Cooperative Institute for Research in the Atmosphere, Colorado State University, Fort Collins, Colorado

Tomi Vukicevic
Department of Atmospheric and Oceanic Sciences, University of Colorado, Boulder Colorado

1. INTRODUCTION

The Regional Atmospheric Modeling and Data Assimilation System (RAMDAS) is a 4 dimensional variational analysis (4DVAR) data assimilation algorithm (Zupanski et. al. 2004) developed at the Cooperative Institute for Research in the Atmosphere (CIRA) at Colorado State University (CSU). The current version of RAMDAS assimilates satellite observations from the Geostationary Operational Environmental Satellite (GOES) and assimilation of other satellite observations are in progress using the same generalized 4DVAR framework. These other satellite observations include the assimilation of the GOES sounder channels and WindSat microwave satellite data (Jones et. al., 2006).

RAMDAS has been used in numerous satellite data assimilation studies (Greenwald et. al. 2002,2004, Jones et. al. 2004, Koyama et. al. 2004, Vukicevic et. al. 2004a). The complexity of assimilating satellite observations into a weather forecast model not only requires the expertise of radiance/satellite and numerical weather prediction/data assimilation scientist, it also requires individuals proficient in system (computational and configuration) integration. This work focuses on the computational and configuration experience of the RAMDAS 4DVAR implementation within the CIRA university research environment. Aspects of our experience could be relevant to other 4DVAR research and operational environments (govt., military, private, etc).

2. IMPLEMENTATION

The ultimate goal of the RAMDAS implementation is to have an adaptable, portable, well-documented and user-friendly data assimilation system. This section describes the current implementation of the system, algorithm, development, documentation and training within RAMDAS.

* *Corresponding author address:* Scott Longmore, Cooperative Institute for Research in the Atmosphere, Colorado State University, Fort Collins, CO 80523-1375; e-mail: longmore@cira.colostate.edu

2.1. System

The software components of RAMDAS are a series of 32 compiled programs that are sequentially executed by a set of BASH command scripts. The programs are primarily written in the FORTRAN 77 and 90 programming languages with some parallel processing and output capabilities being implemented in the C programming language. RAMDAS software also utilizes a set of open source, operating system independent software libraries for parallelization and data output. Parallelization is achieved through the Message Passage Interface (http://www-unix.mcs.anl.gov/mpi/) version 1.1. The Hierarchal Data Format (HDF) library version 5 is currently being implemented for RAMDAS output (http://www.hdfgroup.org/). This format allows easy transportability and is supported by popular visualization analysis and manipulation packages such as IDL and MatLab.

The compilation of RAMDAS programs currently utilizes the Portland Group Fortran (http://www.pgroup.com/) and GNU C (http://gcc.gnu.org/) compilers. The compilation of RAMDAS is accomplished through a set of hierarchal makefiles that utilizes the GNU make program found on most UNIX/LINUX type systems. These makefiles can build the entire system as well as individual executables depending on the command line arguments that are passed at evocation. Another feature of the RAMDAS makefiles is a self-documenting option, which uses the Understanding Fortran package. This feature will be discussed later in the documentation section (2.4).

RAMDAS is implemented on an Intel processor-based High Performance Computing (HPC) 18-node (2 master, 16 nodes) cluster using the SuSE Linux operating system and is connected through a gigabit Ethernet switch (Figure 1). The master node is a dual processor 3Ghz Intel Xeon with 8GB RAM, 1MB cache and hyper-threading. Each node is a 3Ghz Pentium 4 Intel processor 8GB RAM, 1MB cache and hyper-threading. Storage is facilitated through 7500RPM SATA drives configured at RAID5 using a 3ware 9550S-12MI RAID controller. The RAMS forward model component of RAMDAS utilizing 9 nodes completes 1 hour of model simulation time in approximately 15 minutes real clock time, for a typical experiment.

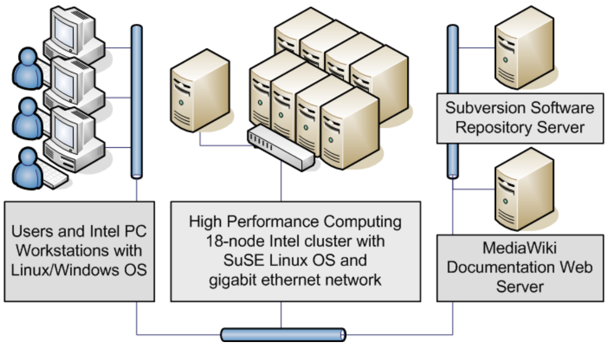

Figure 1. RAMDAS Hardware implementation.

2.2. Algorithm

A scientific discussion of the RAMDAS 4DVAR algorithm has been discussed in Zupanski et. al. (2002, 2004). The focus of this section is to detail the software implementation of the 4DVAR algorithm. The RAMDAS executables can be divided into 3 categories: data preparation, system preparation, and simulation system. The later category, the simulation system, contains the major software components within RAMDAS and is described in Table 1. Each of these categories and major components is a combination of one or more executables that comprise the RAMDAS software. Executables that require a significant amount of computational resources (e.g. the forward and adjoint models and the observational operator) are invoked through the MPI within there respective scripts.

RAMDAS Major Components: 1. RAMS - mesoscale and cloud resolving atmospheric model with coupled land surface model (Cotton et. al., 2003) 2. RAMS Adjoint - The adjoint of a tangent linear version of RAMS (Vukicevic et. al., 2004a) 3. Rtmod - Observational operator for visible and infrared satellite observations. This operator currently uses GOES. (Greenwald et. al., 2002, 2004; Vukicevic et. al., 2004a) 4. Cost function minimization algorithm (Zupanski et. al., 2004)

Table 1. RAMDAS software components.

RAMDAS program execution is illustrated in the sequence diagrams shown in Figure 2. Figure 2a shows the main execution sequence in the run4var script. This script invokes the envir script that sets up environment variables as well as the prep4d script, which runs a set of executables that create initial conditions, computes direct invertible variance, and initializes data assimilation constants and coefficients. Once the prep4d script completes the run4dvar script invokes the cycle4d script that starts the master minimization loop.

The number of times the minimization loop iterates is configurable in the script. Experience from previous experiments shows that 10 minimization iterations is sufficient for convergence on a solution. The sequence diagrams in Figures 2a and b illustrate the cycle4d scripts implementation of the four major components described in Table 1. Figure 2a shows the sequence for the initial forecast guess, incorporation of observations, (in our current configuration, satellite observations), the calculation of sensitivities to the observations through the adjoint model, and the calculation of the cost function. Figure 2b shows the execution of the forecast model on the adjusted initial conditions, the comparison to the observations and the calculation of the cost function alpha for the next minimization step. Also show in Figure 2b is the final forecast from the "best" initial conditions the minimization could provide.

2.3. Development

RAMDAS software development utilizes a set of software utilities and established conventions to ease software debugging and ensure modification traceability and version management. System software versions are archived in the Subversion software version management system (http://subversion.tigris.org/). Multiple users are able to check out individual copies of the entire directory structure and work and make changes concurrently. The development team has implemented the following development strategy for improvements to RAMDAS:

1) User checkout and modification of local copy of RAMDAS source code,
2) User local copy modification testing,
3) Transfer and merge of users modified source code into RAMDAS administrator's local copy,
4) Testing of merged code, and
5) Submission to the RAMDAS repository.

This development strategy allows executable, component, and system testing before releasing into the software repository to prevent bugs being introduced to stable software versions. This strategy also focuses developers on enhancing software and not understanding the intricacies of updating the software repository, which is deferred to the repository administrator.

The RAMDAS development team also utilizes the Totalview (http://www.etnus.com) software debugging system. Totalview is a multi-platform, multi-processor graphical user interface based software analysis and debugging tool. It contains a wide assortment of debugging features including variable and function tracing, scalar/array value graphical representation and memory debugging. The Totalview analysis and debugging software package has helped identify and resolve issues in days to weeks that probably would

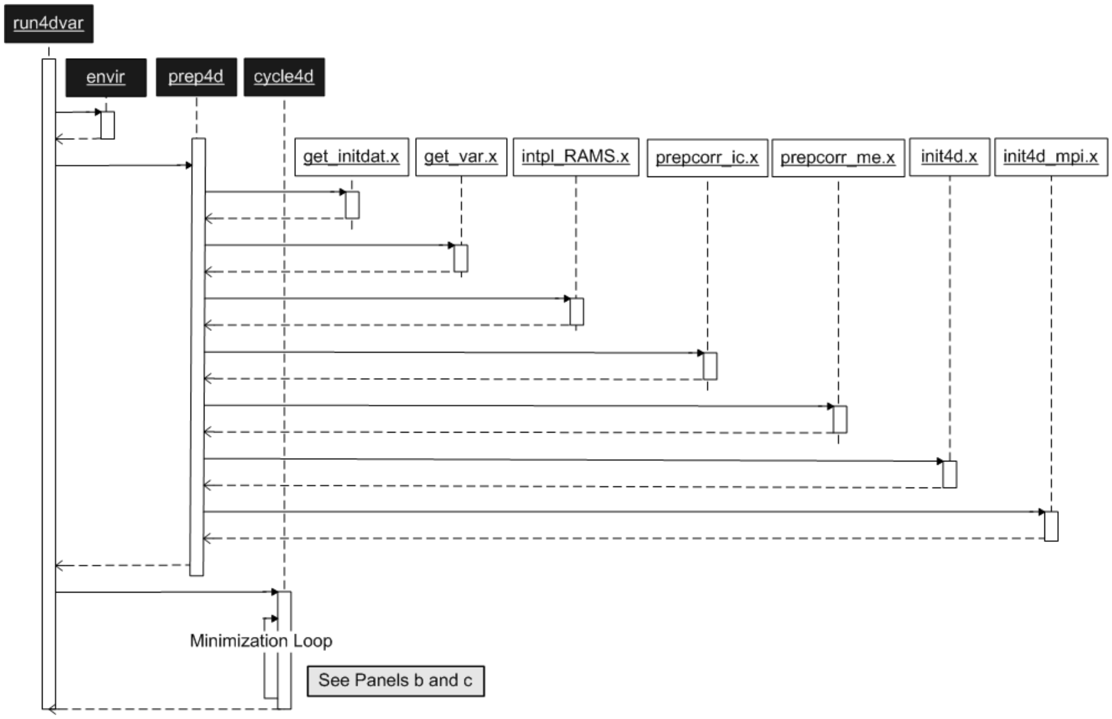

Figure 2a. RAMDAS Master Sequence Diagram. BASH scripts are show in black, executables in white. Scripts and programs are invoked from top to bottom.

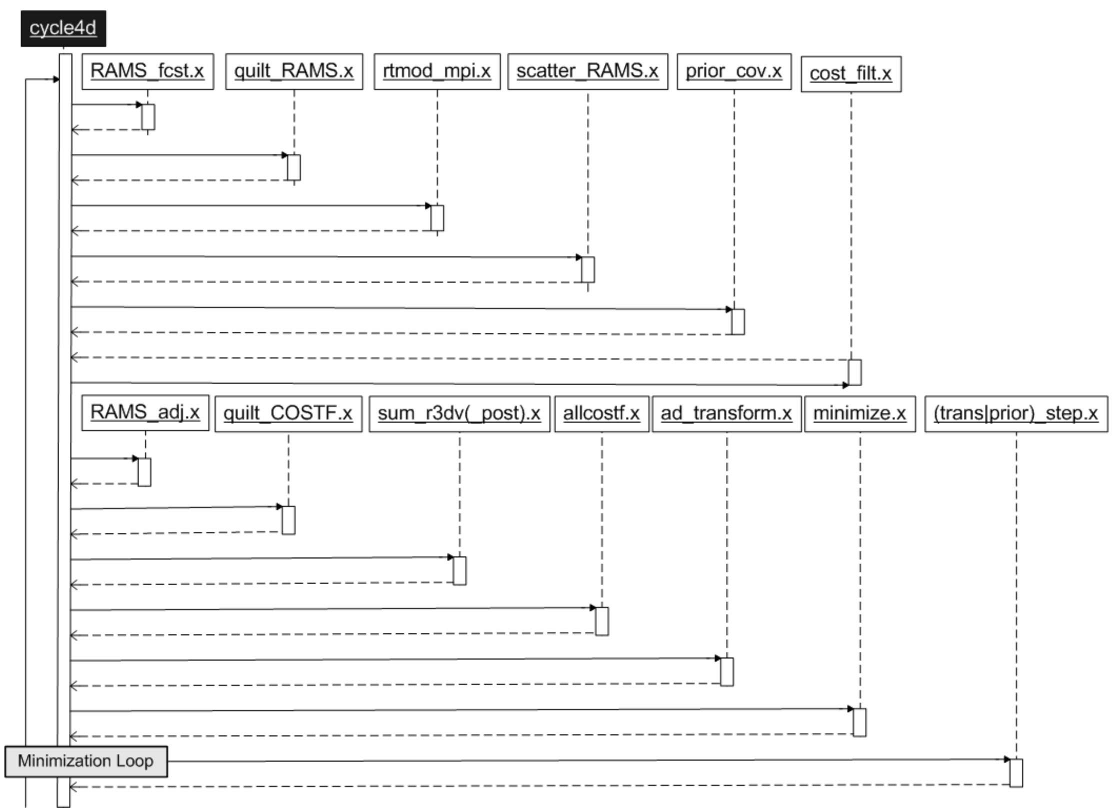

Figure 2b. Same as Fig. 2a except for the RAMDAS cycle4d invocation (initial forecast guess, observational operator and adjoint forecast).

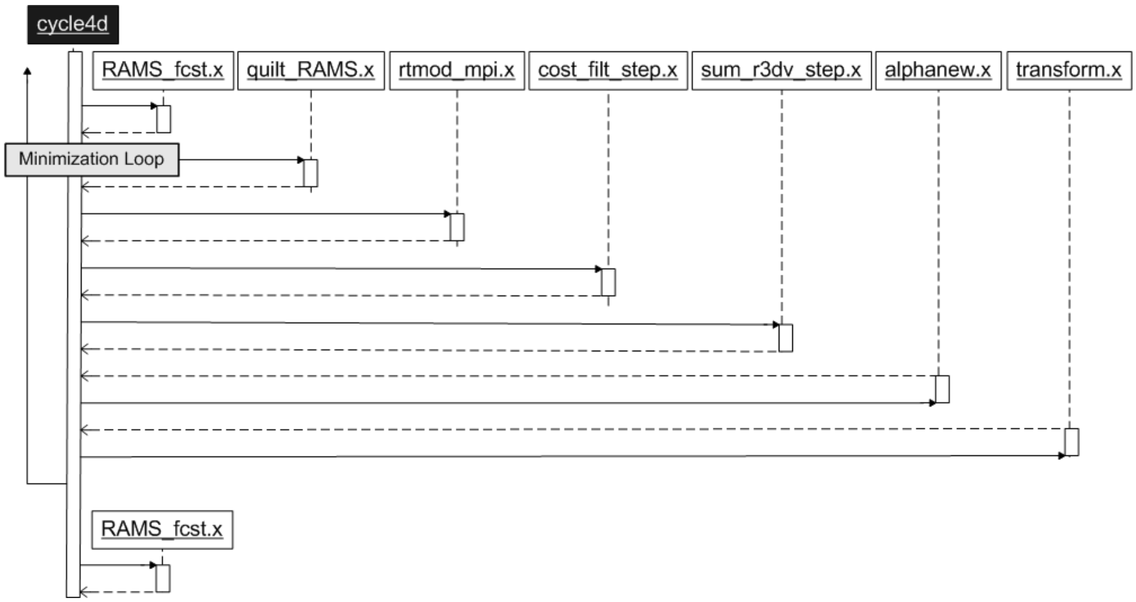

Figure 2c. Same as Fig. 2a except for the RAMDAS cycle4d invocation (step forecast, observational operator and minimization)

have taken months with out the parallel debugging tool. While the cost is significant, the amount of time saved resolving issues is well worth the cost.

2.4. Documentation

Another important component of the RAMDAS implementation is documentation of the system that is easily accessible and quickly amendable as the system evolves. For RAMDAS, documentation has been implemented through MediaWiki, a popular web interface that allows users to add and modify content. MediaWiki as well as other wikis adopt a set of formatting conventions that allow users to quickly organize documentation centrally in multiple web pages. Figure 3 displays the master RAMDAS documentation web page which includes sub web pages such as: System Details, Directory Layout, Relevant Literature, RAMDAS Usage Steps, Current Development, Version Info, Bug Fixes and Log, Executable issues, etc.
All of this content is amendable by RAMDAS users along with the ability to add new sub web pages.

Another documentation utility mentioned earlier is the Understanding Fortran software package. Using a designated option within the program makefiles, a user can create a set of html files that describe and report various aspects and statistics of the RAMDAS software. These html files can then be navigated with a web browser to look at aspects such as an invocation tree report, which shows how nested subroutines are called, or code line statistics in the File Metrics report. Figure 4 shows a sample Understanding Fortran web report. This tool is not only useful for users learning the system; it is also useful as a companion tool for debugging the system.

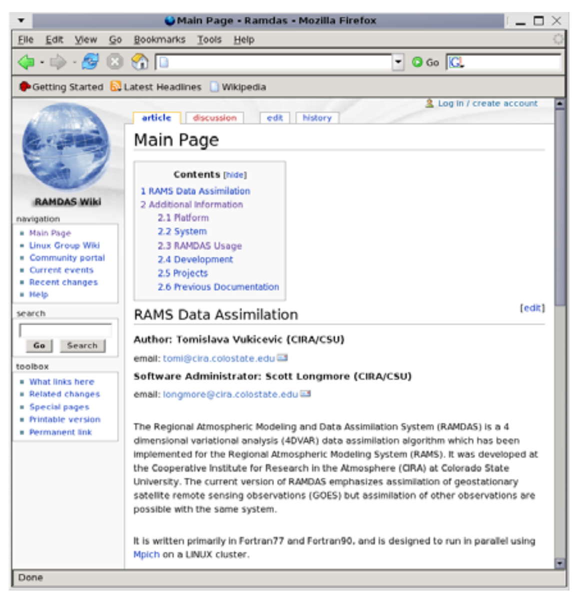

Figure 3. RAMDAS Wiki Documentation

2.5. Training

With the complexity of data assimilation and RAMDAS, system training is crucial for user understanding and usability. One training session has been completed for RAMDAS users and students. The training session utilized a step by step approach for an example case documented on the RAMDAS wiki. After each step was discussed, the step was performed via a live session in real-time. After training completion, users were encouraged to complete the example case themselves as documented on the wiki. Feedback from users was used to improve the case example documentation and for future training sessions.

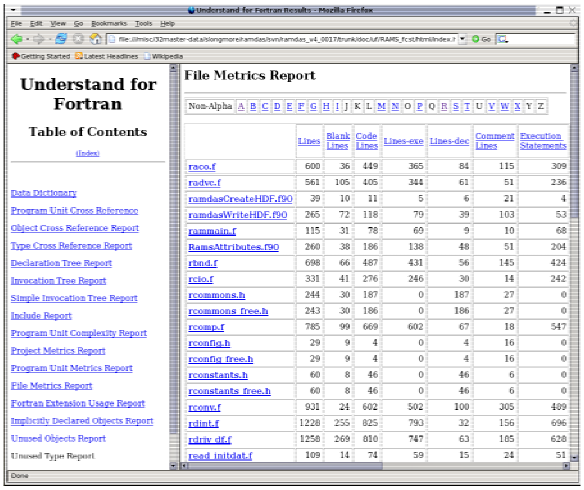

Figure 4. Understanding Fortran software documentation web interface.

3. EXPERIMENTS

Conducting a data assimilation experiment with RAMDAS requires two preparation steps: pre-processing of initial model data and acquiring an assimilation data set and developing corresponding observational operator software (Figure 5). The acquisition of new observation data and the development of a new observational operator is beyond the scope of this paper, but will be included in future documentation of RAMDAS.

There are 5 core steps in pre-processing initial data for a RAMDAS satellite data assimilation experiment. The first step is retrieving global initialization data. For our current experiments we have used the National Oceanic and Atmospheric Administrations (NOAA) Air Resource Laboratory's (ARL) archived National Center for Environmental Prediction (NCEP) Global Data Assimilation System (GDAS) Final (FNL) run data. (URL: http://www.arl.noaa.gov/ready-bin/noaaserver1.pl?metdata=FNL). Once the desired experiment's GDAS FNL data is retrieved, the ARL formatted data can be converted to RAMDAS software utility prep_fnl (Figure 6). The next two steps involve running the RAMS software to create topography and variable files by setting up a set of RAMS configuration files and running the run_prep_all script (Figure 6). Currently for each day's worth of FNL data, steps 2-4 will need to be repeated. The last step involves running RAMS in initialization "spin up" mode, which will create a "history" file that will seed the RAMDAS data assimilation run. Once the history data files exist, modification of RAMDAS configuration files is needed to conduct a data assimilation experiment.

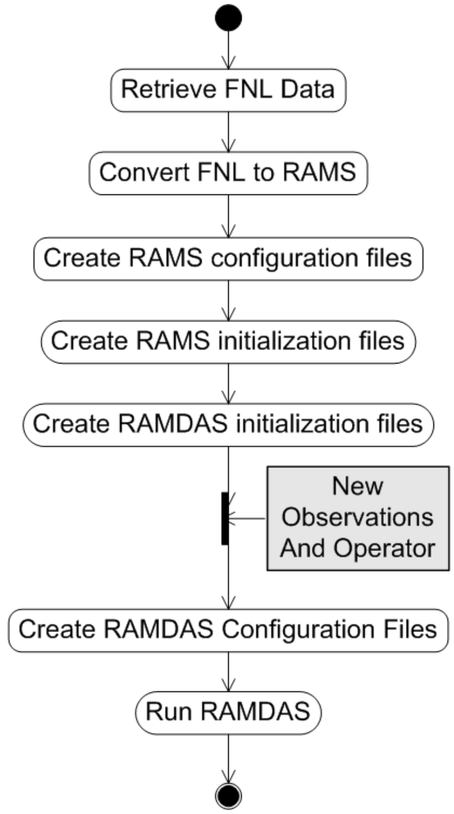

Figure 5. RAMDAS experiment preparation flowchart.

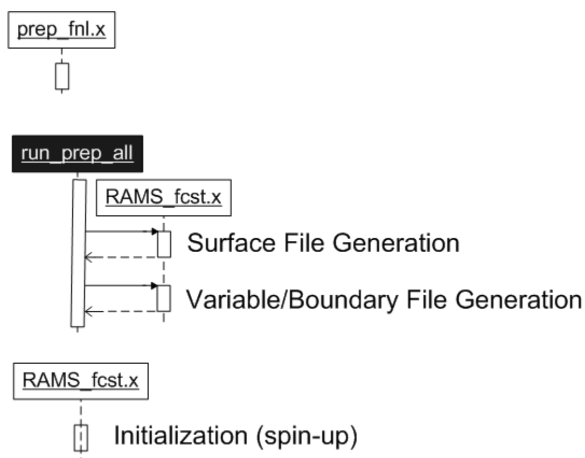

Figure 6. RAMDAS preparation software sequence.

4. EXPERIENCE AND LESSONS LEARNED

Two major themes have surfaced in understanding and enhancing the computational and configuration aspects of RAMDAS; these are identifying the right tools that will benefit the development and understanding of RAMDAS, and identifying the right processes and conventions that will benefit the development of RAMDAS. The complexity of RAMDAS required identifying tools (such as the Subversion, Totalview, Understanding Fortran and MediaWiki) that would quickly streamline the improvement and development of RAMDAS. Establishing processes such as the RAMDAS software development cycle and conventions such as documenting executable issues on the RAMDAS wiki have improved responsibility designation and communication between developers. Identifying tools, processes and conventions has been key to establishing a foundation in which improvements and enhancements can be made to RAMDAS.

5. CONCLUSIONS AND FUTURE WORK

RAMDAS is a 4 dimensional variational analysis (4DVAR) data assimilation algorithm implemented in an intricate collection of software and hardware platforms that requires supplemental software tools, processes, conventions, documentation and training administrated by proficient system (computational and configuration) integration expert(s). These integration experts can act as a liaison between satellite and modeling scientist for this complex, multi-discipline system.

Now that a stable development foundation for improvement and development has occurred for RAMDAS, future work will include optimizing and streamlining software at the system, component and executable software levels, the incorporation of new observations and the development of respective operators, and integration work between the WRF 3D/4D VAR systems. Another aspect of future work is that in education and training in data assimilation techniques similar to that outlined by Vukicevik et. al. 2004b.

REFERENCES

Cotton, W.R., R.A.S. Pielke, R.L. Walko, G.E. Liston, C.J. Tremback, H. Jiang, R.L. McAnelly, J.Y. Harrington, M.E. Nicholls, G.G. Carrio, and J.P. McFadden, 2003: RAMS 2001: Current status and future directions. *Met. Atmos. Phys*., **82**, 1-4 5-29.

Greenwald, T.J., R. Hertenstein, and T. Vukicevic, 2002: An All-Weather Observational Operator for Radiance Data Assimilation with Mesoscale Forecast Models. *Mon. Wea. Rev*., **130**, 7 1882-1897.

Greenwald, T.J., T. Vukicevic, L.D. Grasso and T. Vonder Haar, 2004: Adjoint sensitivity analysis of an observational operator for visible and infrared cloudy-sky radiance assimilation. *Quart. J. Roy. Met. Soc.*, **130**, 597B 685-705.

Jones, A.S., T. Vukicevic, and T. Vonder Haar, 2004: A Microwave Satellite Observational Operator for Variational Data Assimilation of Soil Moisture. *J. Hydromet*., **5**, 1 213-229.

Jones, A. S., C. L. Combs, S. Longmore, T. Lakhankar, G. Mason, G. McWilliams, M. Mungiole, D. Rapp, T. Vonder Haar, and T. Vukicevic, 2007: NPOESS soil moisture satellite data assimilation research using WindSat data, *Third Symposium on Future National Operational Environmental Satellite Systems-Strengthening Our Understanding of Weather and Climate*, January 16-17, San Antonio, TX, submitted.

Koyama, T., T. Vukicevic, M. Sengupta, T. Vonder Haar and A.S. Jones, 2006: Analysis of information content of IR sounding radiances in cloudy condtions. *Mon. Wea. Rev*., accepted

Vukicevic, T., T. Greenwald, M. Zupanski, D. Zupanski, T. Vonder Haar, and A.S. Jones, 2004a: Mesoscale Cloud State Estimation from Visible and Infrared Satellite Radiances. *Mon. Wea. Rev.*, **132**, 12 3066-3077.

Vukicevic, T., E. Kalnay, and T. Vonder Haar, 2004b: The Need for a National Data Assimilation Program. *BAMS*, **85**, 1 48-49.

Zupanski, M. 1996: A preconditioning algorithm for four-dimensional variational data assimilation. *Mon. Wea. Rev.* **124**, 11 2562-2573.

Zupanski, M., D. Zupanski, T. Vukicevic, K. Eis, and T. Vonder Haar, 2004: CIRA/CSU Four-Dimensional Variational Data Assimilation System. *Mon. Wea. Rev.*, **133**, 4 829-843.

ACKNOWLEDGMENTS

This work was supported by the DoD Center for Geosciences/Atmospheric Research at Colorado State University under the Cooperative Agreements DAAD19-02-2-0005 and W911NF-06-2-0015 with the Army Research Laboratory, and by the NPOESS Inter Governmental Studies Program (IGSP).