



U.S. DEPARTMENT OF
ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

The Aerosol Modeling Testbed: Running the Analysis Toolkit Software

William Gustafson Jr., Elaine Chapman, and Jerome Fast

July 2009



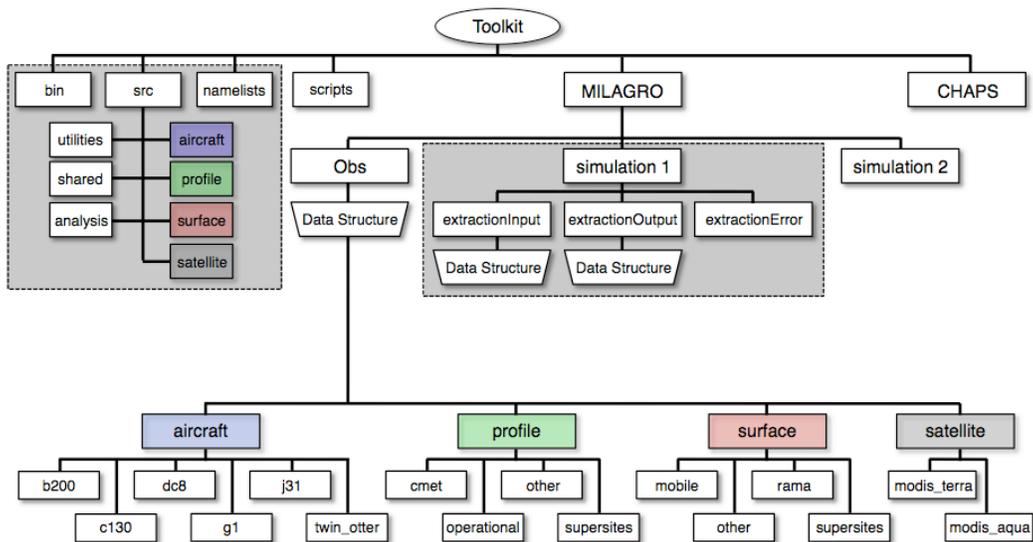
Pacific Northwest
NATIONAL LABORATORY

Aerosol Modeling Testbed: Running the Analysis Toolkit Software

William Gustafson Jr., Elaine Chapman, and Jerome Fast
Pacific Northwest National Laboratory

June 11, 2009
Version 1.0

<http://www.pnl.gov/atmospheric/research/aci/amt/index.stm>
Contact: Jerome.Fast@pnl.gov



1.0 How to use the Analysis Toolkit

This section describes how to run Analysis Toolkit software that extracts the appropriate information from the WRF output files, plots observed and simulated quantities, and computes various statistical measures of model performance. The software is designed to run on a Linux operating system using a Fortran compiler, NetCDF libraries, Perl, and gnuplot. The following is a step-by-step instructions that use the MILAGRO dataset as an example.

Step 1: Obtain the Software and Dataset

First, download the software file (amt_toolkit.tar) and type “tar -xvf amt_toolkit.tar” to extract the folders and files. Then, download the MILAGRO dataset file (data.tar.gz), type “gunzip data.tar.gz”, and “tar -xvf data.tar”. Move the MILAGRO directories into the amt_toolkit/trunk directory.

Note that if you are using the Toolkit on a Lustre based file system, such as on one of the large DOE super computers, you should change the default stripe settings to improve performance of the Toolkit. Because the Toolkit consists of a large number of small files, the typical default

stripe count of two or three does not speed up data transfers and significantly slows down the extraction process due to extra overhead when opening and closing the files. The command to change the defaults for a directory are:

```
lfs setstripe <directory or file name> <stripe size> <stripe offset> <stripe count>
```

The recommended settings for these options when using the Toolkit are:

Stripe size	0	(system default, typically 1 MB)
Stripe offset	-1	(system default, typically round-robin)
Stripe count	1	(Do not split the file onto multiple “OSTs”)

Because changes to the stripe settings only take affect for newly created files, users must change these settings before placing the Toolkit files within a new directory structure. The easiest way to do this is to create a directory to contain the Toolkit, change the stripe settings, and then untar the files into the new directory as described above. To confirm that this has been done correctly use the command:

```
lfs getstripe <directory name>
```

which will show the current stripe settings for a given directory and the files inside it. Because different Lustre configurations are tuned for different purposes, users may want to do tests to confirm that the recommended defaults are efficient for their computer.

The /trunk/MILAGRO/Obs directory is divided into *aircraft*, *profile*, or *surface* directories (the *satellite* directory will be added at a later date). The *aircraft* data is divided into: /b200, /c130, /dc8, /g1, /j31, and /twin_otter. The c130 directory contains data for both 60-s and 10-s averaged data. The 60-s and 10-s averaged data may be more appropriate to compare with coarse and fine WRF domains, respectively. However, NCAR provided more data in the 10-s files than in the 60-s files. The data for the dc8 directory is 60-s averages and the data for the g1 directory is 10-s averages. The *profile* data is divided into /cmet, /operational, /other, and /supersites. The *surface* data is broken up into: /mobile, /other, /rama/, and /supersites.

In each directory, one variable is contained in each file. The file name and the contents in the file should be self-explanatory. A detailed description of the data is given in “Analysis Modeling Testbed: MILAGRO Field Campaign Data in the Analysis Toolkit.” Data is still being added, so the /trunk/MILAGRO/Obs directory should not be viewed as complete yet.

Step 2: Compile the Toolkit Software

- Go to /amt_toolkit/trunk/src
- Edit configure.toolkit for compiler and compiler flags for your system
- netCDF environment variable should be defined for its path, the same as done for compiling WRF
- Go to /amt_toolkit/trunk/src, and type make

Step 3: Extract WRF Variables

- Go to /amt_toolkit/trunk/scripts and edit the processWRF.pl file

- Change \$amtdir to indicate the location of ../trunk
- Change "\$runname to create a unique name of your run. This will become a directory name under trunk as "trunk/MILAGRO/\$runname". This way you can maintain sets of extracted variables for multiple runs
- Change \$wrfoutpath to indicate where the wrfout* files are located for this run
- Change \$wrfdomain to indicate which domain number to extract from
- Change \$incrementFlag to be either hourly or half-hourly to reflect increment of wrfout* files
- Change \$aer_style to indicate either modal (MADE/SORGAM) or sectional (MOSAIC) aerosols used in the wrfout* files
- If running processWRF.pl for the first time, either \$locateAircraftPts or \$locateSurfacePts must be set to 1. Set both to 1 if you want to extract both aircraft and surface data. The locate programs determine how to map the observations to specific grid points in your domain. They only need to be run once per model run.
- The next 10 \$pull* names will extract variables by type. To extract all information, then set all to 1. Extracting all information make take some time (several hours using a single processor), so one can extract one at a time. For example, \$pullAircraftGases could be set to 1 and all others set to zero, and that will cause only trace gases for aircraft flights to be extracted.
- Do not modify anything below "MAIN PROGRAM START"
- Run processWRF.pl . This will extract variables from the wrfout* files and put them in a new directory created automatically in /trunk/MILAGRO/\$runname.

Step 4: Plotting

There are 5 scripts set up to plot various observed and predicted quantities including:

- 1) plot_timeseries_ac.pl – timeseries plots of observed and predicted quantities for aircraft flights
- 2) plot_scatter_ac.pl – scatter plots of observed and predicted quantities for aircraft flights
- 3) plot_scatter_sf.pl – scatter plots of observed and predicted quantities for surface sites
- 4) plot_percentile_box_whisker_ac.pl – mean, 95th percentile and range plots for observed and predicted quantities for aircraft flights

For each script, edit the following line:

- Change \$exp to the path of the /MILAGRO directory
- Change \$run to the directory name containing WRF output from the processWRF.pl script
- Change \$aerosol_setup to either 4binWRF or 8binWRF for MOSAIC
- gnuplot v4.2 or greater (freeware) is assumed to be installed at /usr/local/bin/gnuplot. If it is located somewhere else, the path of \$gnuplot needs to be changed.

For the aircraft scripts, the flights.txt file can be edited to include only those aircraft flights to be analyzed. By default it is set to all of the available flights.

For the surface scripts, the categories.txt file can be edited to include only those types of surface sites to be analyzed. By default it is set to all available categories

The plot*.pl scripts can be run one-at-a-time. In the future, these scripts may be merged into one script. Output in the form of postscript files (*.ps) will be produced with the filename based on the variable and aircraft flight or surface site. Note that some system errors may be generated even when everything is working, so don't worry unless the results are missing.

Step 5: Statistics

There are 2 scripts set up to compute statistics including:

- 1) get_stats_ac.pl – various statistical measures of model performance for aircraft flights
- 2) get_stats_sf.pl – various statistical measures of model performance for surface sites

For each script, edit the \$exp, \$run, and \$aerosol_setup lines, similar to the plotting scripts.

The output is text that should be directed to a file to save the results (i.e. ./get_stats_ac.pl >file.out). The format is tabular and can be pasted into Word or Excel. Note that some system errors may be generated even when everything is working, so don't worry unless the results are missing.

2.0 Using the Satellite Portion of the AMT Toolkit

There are three major steps within the AMT Toolkit to extract concurrent observation and model data for satellites. The first step is to define the comparison grid. The second step is to regrid the satellite data to the comparison grid. And, the third step is to do the same thing for the WRF output. In some cases, the grid may be identical between the input and output, such as when matching to a pre-existing WRF grid.

Step 1: Defining the Comparison Grid

The purpose of this step is to setup a grid that will be used for comparing WRF output with satellite data. Depending on the purpose, e.g. quick-look plots versus rigorous statistics, a different grid may be desired. The grid location can also either be static or time dependent.

Currently, no automation scripts exist for generating the comparison grids. So, the first task is to ensure that a suitable output directory exists for the grid files. A recommended location is to create a directory called "outgrids" at the same level as the Obs directory. For the MILAGRO case, this would be ../amt_toolkit/MILAGRO/outgrids.

The user defined settings to generate a comparison grid are located in the definegrid.nl namelist file. Copy the default definegrid.nl file from the namelists directory to the newly created outgrids directory. Then, edit the new copy as needed for the desired grid. The namelist contains a number of different blocks that may or may not be needed depending on the type of grid chosen with define_method.

grid_choice block: General choices used by all grid types.

define_method:

- 0 WRF_GRID, copies a grid from a WRF domain that is static in time
- 1 SAT_GRID, generates a time dependent grid from a series of satellite files (not yet supported)
- 2 USER_GRID, generates a cylindrical equidistant (lat-lon) grid defined by the user

grid_name: The name of the comparison grid file. The “nc” extension is not included in the name.

wrf_grid_params block: Choices specific to the WRF_GRID type.

wrfout: Name of a wrfout file from which to get the lat-lon coordinates.

i1, i2, j1, j2: These parameters set the specific grid points that should be used within a given WRF domain. If the numbers are negative, that signifies the number of points away from the boundary edge where the comparison grid begins. For example, if i1=i2=i3=i4=-6, the outer five ring of points around the domain edge are skipped and the comparison grid would begin at the 6th point in from the edge. If these variables are set to positive values, then that defines the absolute grid point number to use for the edge of the comparison grid. The “i” variables are for the W-E direction and the “j” variables are for the S-N direction.

sat_grid_params block: Choices specific to the SAT_GRID type. This is not yet used.

user_grid_params block: Choices specific to the USER_GRID type.

lat_sw: Latitude of southwest grid point

lon_sw: Longitude of southwest grid point

lat_ne: Latitude of northeast grid point

lon_ne: Longitude of northeast grid point

dlon: Longitude grid spacing in degrees

dlat: Latitude grid spacing in degrees

Note that if the distance between the SW and NE directions are not evenly divisible by the grid spacing, the northeast corner is adjusted a bit.

Once the choices have been made in the namelist file. The grid is then generated by running the definegrid.exe program. Definegrid.exe must currently be located in the same directory where definegrid.nl is located.

Step 2: Regridding the Satellite Files to the Comparison Grid

Currently, only MODIS aerosol and cloud data is setup for regridding, so this description assumes that this is the type of data being regridded. Regridding the satellite files involves matching the satellite swaths to the comparison grid, generated in Step 1, and to a desired set of times. First, a directory should be created to hold the output. If the processWRF.pl script has already been run for this dataset, a set of satellite directories will have been created for the given

run. It is recommended to place the regridded satellite data into the appropriate directory within a given run as there is typically a one-to-one correspondence between a set of regridded satellite and WRF data. Once a directory has been chosen, copy the pullvar_st_modis.nl namelist from the namelists directory into the desired output directory.

The choices within the pullvar_st_modis.nl file are:

- gridfile: Name of the grid file (the comparison grid created in Step 1)
- outtimefile: Name of a file containing a list of times to output (see below). If set to "From wrfout" a time series is generated based on a series of WRF files.
- datapath: Path to the satellite files, typically within the Obs/satellite/... structure
- category: The first handful of characters of the satellite input filenames. This is used to determine which files in the directory to include in the output. For example, MY06_L2 to get the MODIS cloud products from Aqua.
- minconfidence: Minimum allowed confidence level of subsetted data. For MODIS, the allowable values range from 0 to 3, where 3 is the highest confidence.
- timemethod: 0 = Use the instantaneous satellite time closest to the output time
1 = Average the satellite times that reside within the output time block
- spacemethod: 0 = Use nearest neighbor interpolation
1 = Average input grid cells whose center lie within the output grid cell. This is only appropriate when the input grid has higher resolution than the output grid.
- searchrad: Minimum radius to search for nearest neighbors in units of meters. This is only used for spacemethod=0. If negative, then the default of 3 output grid boxes is used.
- numvar: Number of output variables
- namevar: List of variables to output. The names must correspond exactly with the variable names hard-coded into the pullvar_st_modis.exe program. Also, all variables processed at one time must be of the same input resolution, e.g. 1-km and 5-km cloud products must not be processed simultaneously.
- outfile: Name of the output file
- wrfpath: Path to a list of WRF files (only used with outtimes="From wrfout")
- wrfdomain: Domain number of WRF files to use (only used with outtimes="From wrfout")
- incrementFlag: Time increment between WRF files, 1=hourly and 30=half-hourly (only used with outtimes="From wrfout")
- nocolons: Flag indicating if colons (0) or underscores (1) are used in the WRF filenames (only used with outtimes="From wrfout")

If a user-selectable set of output times is used, a text file needs to be created indicating the specific times, as indicated with the outtimefile variable. This file contains two columns, with the first column containing a timestamp and the second column containing the length of the time period (in minutes). Only satellite files residing within the given period are included in the search

for data. This allows one to pull all the satellite swaths for a given day into one output time. For the example shown in Table 1, the six-hour period centered on 19 UTC each day is included in the search. Typically, all MODIS data will lie within a 2 hour period for the region surrounding central Mexico, but the six hour period is used as a buffer in case a third overpass might give more data.

Table 1: Example Output Time File	
2006-03-01_19:00:00	360
2006-03-02_19:00:00	360
2006-03-03_19:00:00	360
2006-03-04_19:00:00	360
2006-03-05_19:00:00	360
2006-03-06_19:00:00	360
2006-03-07_19:00:00	360
2006-03-08_19:00:00	360
2006-03-09_19:00:00	360
2006-03-10_19:00:00	360

Once the namelist is setup, run pullvar_st_modis.exe to extract the satellite data.

Step 3: Regridding WRF Output to the Comparison Grid

Regridding the WRF output to the comparison grid is similar to what was done for the satellite data. Begin by copying the pullvar_st_wrf.nl namelist from the namelists directory into the appropriate output directory.

The settings in pullvar_st_wrf.nl are:

- gridfile: Name of the grid file (the comparison grid created in Step 1)
- outtimefile: Name of a file containing a list of times to output (see above). If set to “From wrfout” a time series is generated based on a series of WRF files.
- timemethod: 0 = Use the instantaneous satellite time closest to the output time
 1 = Average the satellite times that reside within the output time block
- spacemethod: 0 = Use nearest neighbor interpolation
 1 = Average input grid cells whose center lie within the output grid cell. This is only appropriate when the input grid has higher resolution than the output grid.
- searchrad: Minimum radius to search for nearest neighbors in units of meters. This is only used for spacemethod=0. If negative, then the default of 3 output grid boxes is used.
- numvar: Number of output variables
- namevar: List of variables to output. The names must correspond exactly with the variable names hard-coded into the pullvar_st_modis.exe program. Also, all

variables processed at one time must of the same input resolution, e.g. 1-km and 5-km cloud products must not be processed simultaneously.

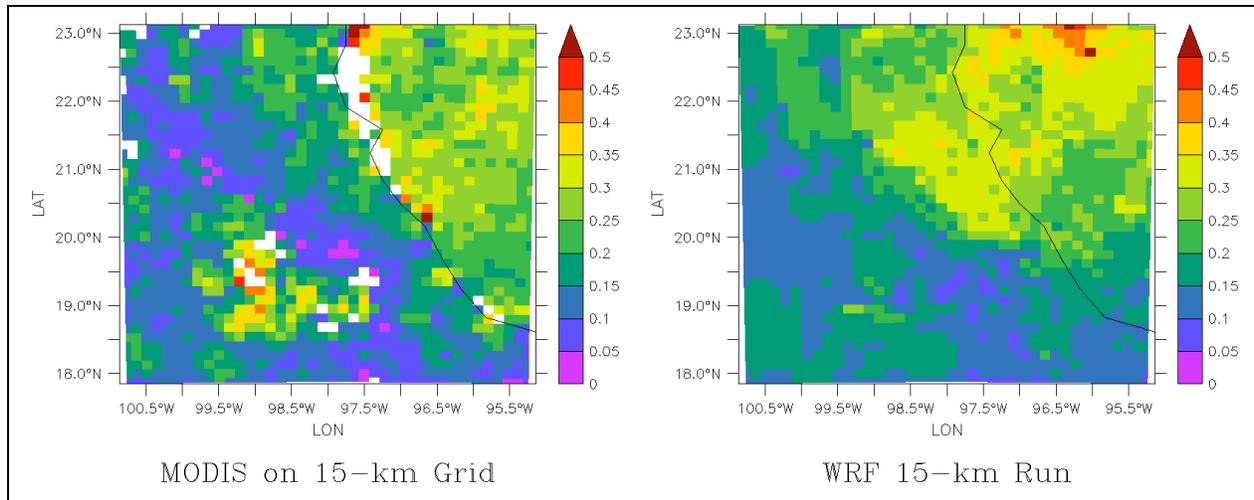
- wrfpath: Path to a list of WRF files (only used with outtims="From wrfout")
- wrfdomain: Domain number of WRF files to use (only used with outtims="From wrfout")
- incrementFlag: Time increment between WRF files, 1=hourly and 30=half-hourly (only used with outtims="From wrfout")
- nocolons: Flag indicating if colons (0) or underscores (1) are used in the WRF filenames (only used with outtims="From wrfout")
- outfile: Name of the output file

Once the namelist is setup, run pullvar_st_wrf.exe to extract the WRF output.

Plotting and analysis:

After completing Steps 1-3 a pair of files has now been generated that can be directly compared because they have points that are coincident in time and space. The file format is netCDF that can readily be used within most common plotting software including Ferret, IDL, Matlab, NCL, etc. For example, a sample Ferret script to plot two times is shown in Table 2.

Table 2: Example Ferret script to plot regridded MODIS and WRF output side-by-side
<pre> !----- ! Example Ferret script to plot MODIS vs WRF Toolkit output. ! William.Gustafson@pnl.gov; 12-Jun-2009 !----- ! Some user choices... let satvar = OPTICAL_DEPTH_LAND_AND_OCEAN let wrfvar = TAU_AER550CM_COLUMN define symbol thelevs = (0,.5,.05) (Inf) ! Open the files... use MYD04_on_middle15kmgrid_minconfidence3.nc use wrf15km_on_middle15kmgrid.nc ! Setup the window... set window/aspect=.5 cancel mode logo ! Average all the days together into a monthly mean... let wrfave = wrfvar[l=@ave] let satave = satvar[l=@ave] ! Make the plots... set v left shade/title="MODIS on 15-km Grid"/lev=(\$thelevs)/d=1 satave, lon, lat go land 1,,1,1 set v right shade/title="WRF 15-km Run"/lev=(\$thelevs)/d=2 wrfave, lon, lat go land 1,,1,1 </pre>
<p>Output from the above script:</p>



3.0 Tailoring How the Analysis Toolkit Functions

For the extraction of WRF variables:

- /src/aircraft: pullvar_ac.nl, pullvar_ac_dircect.nl, and pullvar_acprof.nl
- /src/surface: pullvar_sf.nl

These files control which WRF variables are extracted along aircraft flight paths and at surface sites. Many variables are extracted, but not all possible output is extracted since it may not be comparable to the available data. The easiest way to do this is to edit the master copy of the namelist files in the src directories. The processWRF.pl script then copies these out to the MILAGRO data directories. If you edit a namelist file that has already been copied to a data directory, you can manually run the associated pullvar from within that directory, but the namelist will get overwritten next time processWRF.pl is run.

For plotting and statistics scripts:

- /src/analysis/comparable_files_ac_8binwrf.txt
- /src/analysis/comparable_files_sf_8binwrf.txt

These files are used to determine which observed and simulated files to compare. The names of the observed variables are sometimes duplicative, because they are named differently among organizations or there is more than one measurement of the variable on a particular flight or surface site. For example there are 4 peroxide observations: h2o2_obs, h2o2_cit_obs, h2o2_uri_obs, and h2o2_usna_obs. The programs will compare model output wherever those observations are available

4.0 Directory Structure

A schematic of the directory tree used by the Toolkit is depicted below. There are separate programs in the /src directory to handle aircraft, profile, surface, and satellite data. Data in the /Obs directory is divided into similar categories. When the processWRF.pl script is run, it generates a similar directory structure as in the /Obs directory.

