Summer Institute in Earth Sciences 2009

Comparison of the NASA GEOS-5 Model to MPLNET Aerosol Data

Bryon J. Baumstarck
Departments of Physics, Computer Science, and Mathematics
Rocky Mountain College, Billings, Montana

Mentor: Dr. Peter Colarco
Atmospheric Chemistry and Dynamics Branch
Code 613.3
NASA Goddard Space Flight Center, Greenbelt, Maryland

ABSTRACT

Aerosols are particulates and droplets suspended in the atmosphere, and can come from both natural and anthropogenic sources. They affect Earth's climate by scattering and absorbing solar radiation and modifying the properties of clouds. Despite their importance, there are large uncertainties in how aerosols affect climate because of the lack of observational data. In order to better constrain these uncertainties, scientists use an array of tools to measure aerosols, including satellites, radar, Lidar, and others. In this project, we focused on one particular system of Lidars—which measure the vertical profile of aerosols—called the Micro-pulse Lidar Network (MPLNET). We compared MPLNET data to the representation created by the NASA GEOS-5 model, which simulates the global distributions of aerosols. This project used a baseline plotting procedure I created for both MPLNET data and GEOS-5 data in order to create an accurate visualization of the model, which has never been done before. This is accomplished by using the IDL programming language to create contour plots with consistent leveling parameters. Using this consistent plotting procedure we visually compared the graphs and found little or no correlation between the model and the MPLNET aerosol data.

1. Introduction

Aerosols are particles or liquid droplets suspended in the atmosphere. They can be from natural sources, like dust or sea spray, or from anthropogenic sources, like pollution or biomass burning. These particles interact with the atmosphere in various ways, some to a greater extent then others. Aerosols affect the air you breath and are connected to numerous upper respiratory conditions, namely asthma. Aerosols can even cause global effects such as climate change. The extent to which aerosols play a role in climate change is uncertain, but generally speaking aerosols cool the Earth's surface (IPCC report). Aerosols cool the Earth's surface by either absorbing or scattering solar radiation, and by changing the properties of clouds. However, how and to what extent aerosols interfere with solar radiation and cloud formation is unknown.

To understand this uncertainty better, scientists use a variety of remote sensing instruments to measure the properties and effects of aerosols. These instruments can be satellites, land-based, ocean-based, aircraft, and include Lidar, radar, sun photometers, and others. By understanding how scientists measure the properties and effects of aerosols, one is able to create an accurate visual representation of the data. This visualization is then used to further analyze the data.

In addition, scientists use various models to help analyze and predict trends in the effects of aerosols. One of these models is the Goddard Earth Observing System Model version 5 (GEOS-5). These models are based on algorithms that take into account the dynamics of the whole Earth system, and give a representation of how the aerosols are moving and interacting with the atmosphere. In order to have confidence in the modeled aerosol distributions, we need to evaluate the models in the context of aerosol observational data sets. In this project we compared the GEOS-5 model results to the data from the NASA Micro-pulse Lidar Network (MPLNET), which is a network of ground-based Lidars that measure the vertical profile of aerosols.

Currently there are 34 MPLNET sites with more sites planned, and the locations of these MPLNET sites are shown in Figure 1. This list of sites was then narrowed down to only the sites that were operational during the month of April in 2008, and further narrowed down to a specific site, Goddard Space Flight Center (GSFC). By narrowing the data down to this level, it makes it possible to validate the new method by comparing the new plots with the previous plots done of the MPLNET data. This comparison is key to the validity of the new method because the new method uses a different approach to plotting the data. This new method allows for the comparison of the GEOS-5 model data by providing a consistent plotting program. By maintaining this constant method, one can perform a comparison of the GEOS-5 model to the actual MPLNET data for the purpose of validating the visualization of the model.

2. Instrumentation

a. MPLNET

The Micro-Pulse Lidar Network is a federated network of Micro-Pulse Lidar (MPL) systems designed to measure aerosol and cloud vertical structure continuously. This network is located mostly in Eastern North America and Eastern Asia, with locations scattered throughout South America, Western North America, Africa, the Middle East, the Arctic, and Antarctica. Most MPLNET locations are combined with Aerosol Robotics Network (AERONET) sun photometer locations, and this combination gives the maximum level of data reliability when the MPLNET data is collected during daylight hours. The MPL (see figure 2) is a ground-based, compact, and eye-safe Lidar system capable of determining the range of aerosols and clouds by firing a short pulse of laser light vertically at 523nm and measuring the time-of-flight from pulse transmission to reception of a returned signal. The returned signal is a function of time; it is converted into range using the speed of light, and is proportional to the amount of light backscattered by atmospheric molecules (Rayleigh scattering), aerosols, and clouds (http://mplnet.gsfc.nasa.gov/instruments.html).

b. AERONET

The Aerosol Robotics Network (AERONET) is a global network of ground-based sun photometers. These sun photometers measure the total column amount of light that is extinguished between the sun and the ground. This "extinction" is being used by MPLNET as a constraint on the data to produce various products. (http://aeronet.gsfc.nasa.gov/index.html).

c. GEOS-5 Model

The GEOS-5 model is a general circulation model, and is comprised of multiple modules. The module that is of interest for this paper is the Goddard Chemistry, Aerosol, Radiation, and Transport (GOCART) model, which models the sinks, sources, and transportation of aerosols. The model is a global model that has a 1/2º by 2/3º (approx. 50 km) resolution. (http://geos5.org/wiki/index.php?title=Main_Page#GEOS-5_Systems).

3. Data Description

Data products for MPLNET are seen in 4 levels with each level having one or more possible sublevels. These levels are:

Level 0: Raw Data [RD]
{Real time}
Raw signal counts from the MPL, includes diagnostic parameters.

Level 1: Normalized Relative Backscatter [NRB] Signals (see figure 4)
{Non-quality assured}
Remove all instrument related parameters except the Calibration Constant (C). Shows vertical structure over time, and helpful to access overall vertical structure.

Level 1.5a: Real Time Aerosol Properties (see figures 5, 7, & 9)
{Non-quality assured}

Includes MPL Calibration Value; Aerosol Vertical Structure; Layer Averaged Aerosol Extinction-to-Backscatter Ratio (S); Aerosol Backscatter, Extinction, and Optical Depth Profiles; Cirrus Flag indicating cloud contamination. Coincident MPL and AERONET data is most accurate, but must be during daylight hours due to reliance on sunphotometer.

Level 1.5b: Real Time Aerosol, Cloud, and PBL Heights (see figure 11)

{Non-quality assured}

Includes Planetary Boundary Layer (PBL) heights; Cloud Base and top heights for multiple layers; Altitude of the top of the highest Aerosol layer; Flag indicating beam blockage from thick clouds; Vertical Feature Mask (Aerosol, Cloud, PBL, and combinations). Cloud and Aerosol heights are algorithm based, and PBL heights are generated using a modified Wavelet technique.

Level 1.5c: Real Time Cloud Properties

{Non-quality assured}

DEVELOPMENT. Will contain the optical depth of detected cloud layers, and includes optically thick clouds (Full attenuation of LIDAR).

Level 1.5d: Polar Stratospheric Clouds.

{Non-quality assured}

DEVELOPMENT. Heights and properties of Polar Stratospheric Clouds.

Levels 2.0a,b,c: Quality Assured (1.5a,b,c)

{UNDER CONSTRUCTION}

Level 3: Reprocessed Data

{UNKNOWN, Based off of EOS conventions}

(http://mplnet.gsfc.nasa.gov/data.html).

The Normalized Relative Backscatter, or NRB, is the top-level product available from the MPL instrument (figure 4). The NRB shows an overall picture of the vertical structure of the aerosol in the atmosphere, and is not cloud screen or quality assured. The purpose is to just give a complete picture. The bands of solid white in figure 4 are points in the data set that have missing data. This missing data can be due to an instrument malfunction, movement of the instrument, covering of the signal for the protection of the telescope, or other mechanical reasons. The sections on the figure that are red are areas where there is attenuation of the Lidar signal. This attenuation can be due to rain, heavy low-level clouds, and/or high concentrations of aerosols. However, this product does not give very much detail on the aerosols, but to get this detail we need to have a constraint on the data.

The Aerosol Optical Depth, or AOD, is a measure of how much light is extinguished from a certain altitude to the top of the atmosphere (figure 9). The total column AOD is measured by the AERONET sun photometer, and is then used on the MPLNET NRB value as a constraint to create a Lidar profile of the AOD (figure 3). This column amount is the integration of the AOD from the surface to the top of the atmosphere. We can then use the AOD to get the Backscatter and Extinction of the aerosols.

The aerosol backscatter (figure 5) is like the NRB, however, the backscatter has been corrected for molecular scattering and is cloud screened. This cloud screening data comes from the AERONET sun photometer measurements. If the sun photometer detects clouds, the backscatter value is changed to show no data for that time. For example, in figure 4 between days 111 and 113 there is an area of high low-level attenuation, by comparing this time to true color satellite imagery, it is shown that a think cloud front moved over GSFC. Due to this cloud cover, the backscatter value is removed, thus replacing this area with a band of white in figure 5.

The aerosol extinction (figure 7) is the derivative of the AOD over altitude. This is a measure of how much the aerosol at each layer extinguishes the Lidar signal. This gives a picture of the possible characteristics of the aerosol particles.

4. Method

The method we are using in this project is to use the Interactive Data Language (IDL) programming language to program contour plots of the data. For a complete description of created programs, see the “Read Me” in the Appendix. In order to create the contour plots, we needed to get the actual data from the files. I created a program to read the MPLNET data files, which are in netCDF format. Dr. Welton, the MPLNET Principle Investigator (PI), provided the files used. The reader performs the exclusive functions of opening the selected file, initializing the variables to be used by the plotting program, and closing the file. This simplifies the code tremendously, as you do not have to copy this into each individual plotting program, and it increases the upgradability of the code significantly.

The plotting method we chose to use was to have a different program for each level of product, as described above, and save the plotted graph as an independent file. This division is expanded to three programs in the case of the Level 1.5a files, as there are three independent variables being plotted. This allows for the comparison of the new graph to the old daily graph used on the MPLNET site. The plotting program is also designed to plot a range of files, and this allows for the plotting of MPLNET data for a week, or more, of continuous data. This expansion helps to allow for trend tracking of the aerosols over a particular site so one can see cloud transportation and other phenomenon.

The plotting technique used in these programs is a plot routine provided by Dr. Colarco. That performs dimension checking and includes capability for variable level width, as used in the model data. This technique takes twice as long to compute as the built-in IDL function, but has variable level functionality. More time was needed to improve the efficiency of the plot routine. The contouring level and color scales are identical for the MPLNET plots and the GEOS-5 plots. This duplicity is essential to perform validation of the model. Without this constancy, there would be too much variability in the plotting procedure to give any kind of validation.

The created graphs are saved in Postscript (.ps) format. These postscript files are then converted by using the ImageMagick program's "convert" function to convert the postscript file into a Portable Network Graphic (.png) file. It is these PNG files that are visually compared. Data comparison between the model and actual data would be the next step in determining accuracy of the model.

5. Results

By visually comparing the contour graphs of the GEOS-5 model to the graphs of the MPLNET data, we should see features that are common in both graphs. However, we do not see this correlation. This result comes from the fact that both graphs have identical leveling information – the main focus of performing this project – and that both graphs have the same elevation range. The only other difference between the two graphs is that the GEOS-5 model is on a six-hour time-step. This is a very large difference compared to the MPLNET time-step of one minute. This time difference could be the cause of the discrepancy, as the model uses an average over a much larger time. This average could give a false value to be plotted on the graph; therefore, a precise comparison of the data at each point is needed to verify the model data. This being the case, it is somewhat difficult to match the MPLNET graphs with the GEOS-5 graphs. However, there should still be consistencies between the two graphs. This has not been shown to be the case for all three Level 1.5a variable's graphs (see figures 4, 6, & 8).

6. Conclusion

Since the new method is using the same data as the previous MPLNET plots, and the newly created plots match up to the previous MPLNET plots, we can conclude that the new plotting method is accurate. However, the GEOS-5 model graphs do not match up to the MPLNET graphs. Since we have taken out all variables between the MPLNET graphs and the GEOS-5 graphs, we must conclude that the model is inaccurate. The inaccuracy could be due to an inaccurate algorithm, and this could be due to the uncertainty of how aerosols interact with the atmosphere. Further, this discrepancy could be due to the lack of observational data, as models are based on observed patterns. More information is needed to solve this discrepancy.

Future areas of focus on this project include: first, meticulous comparison of the data in the GEOS-5 model to the actual data at each point is needed to determine where the difference in the two graphs is located. Second, one needs to look at the comparison of the AERONET sun photometer coincident times with the GEOS-5 model, as the model might be more closely related to these values then to the MPLNET data. Third, one needs to look at a regression line comparison of the data to see by how much the model is incorrect. Fourth, one needs to look at expanding the plots of the MPLNET data to include multiple sites. This would allow for accurate measurement of long distance aerosol transport by looking at the aerosol information at each site in panorama. This would allow for determination of how far aerosols travel, in addition to how the higher-level aerosols interact with lower-level aerosols.

Acknowledgments

A huge thank you goes out to my mentor, Dr. Peter Colarco, for his support and willingness to take me on for this project, as well as putting up with my many questions. A special thank you goes to the MPLNET Principle Investigator, Dr. Ellsworth J. Welton, for his insight and help with locating the data. A huge thank you goes to Valerie Casasanto for coordinating the internship program and answering my various questions. A grateful thanks to all my professors at Rocky Mountain College for their demand of academic excellence; especially my Physics professor, Toby Anderson, who pointed me to this internship program at the beginning.

An enormous thank you to NASA for this amazing opportunity, which was provided through the National Aeronautics and Space Administration's Goddard Earth Sciences and Technology Center's Summer Institute in Earth Sciences program.

REFERENCES

Holben, Brent, et. el, Principle Investigator for AERONET. http://aeronet.gsfc.nasa.gov/index.html. Information retrieved June 1 through August 7, 2009.

NASA Center for AeroSpace Information, 2005: *NASA Laboratory For Atmospheres 2005 Instrument Systems Report*, 24, http://atmospheres.gsfc.nasa.gov/pdf/InstrumentReport212783_2005.pdf. Retrieved June 2009.

Solomon, S., Coauthors, 2007: Climate Change 2007: The Physical Science Basis, *Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change*, Cambridge University Press, 14.

Welton, Ellsworth J., Coauthors, 1999: Ground-based Lidar measurements of aerosols during ACE-2: instrument description, results, and comparisons with other ground-based and airborne measurements.

Welton, Ellsworth J., et. el, Principle Investigator for MPLNET. http://mplnet.gsfc.nasa.gov/index.html. Information retrieved June 1 through August 7, 2009.

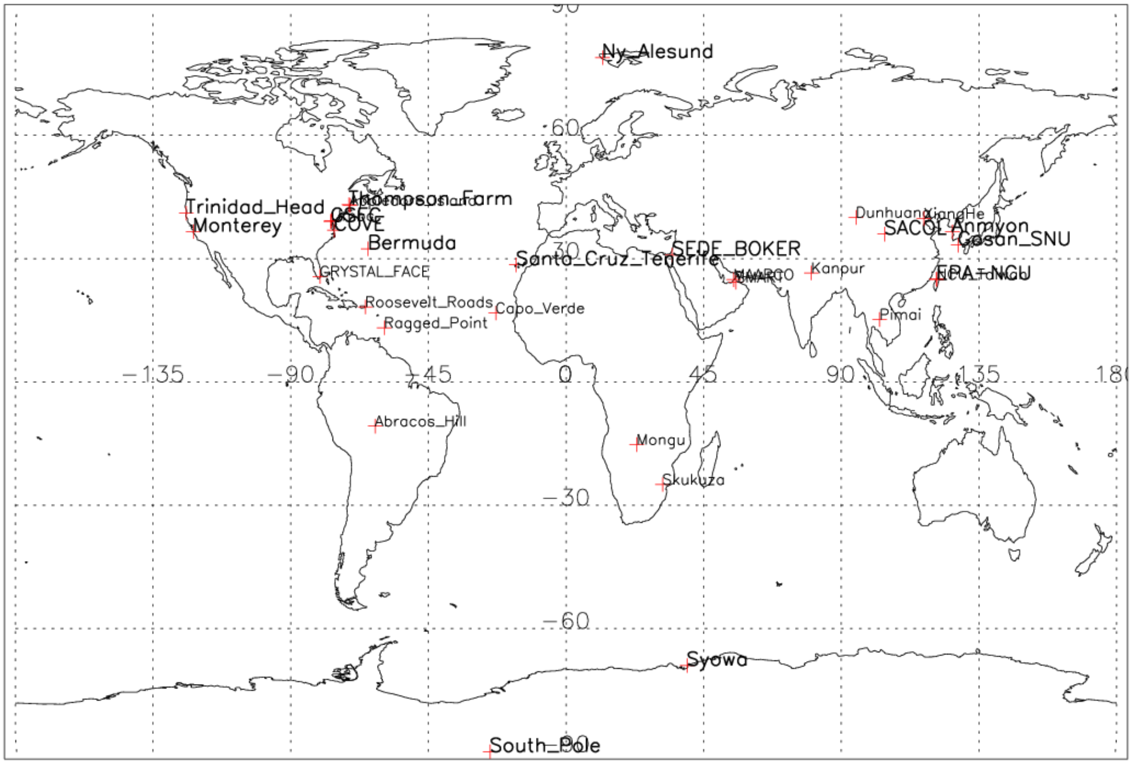
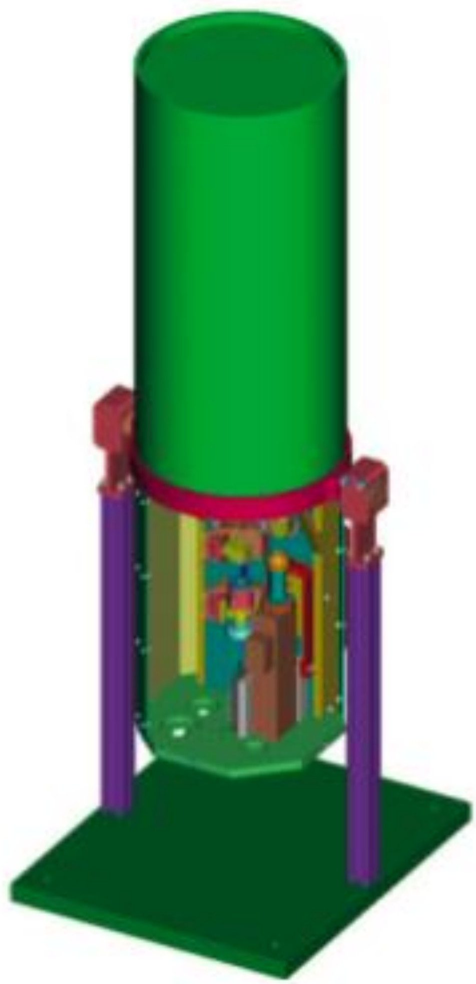


Figure 1: Map of MPLNET site locations. Map created using IDL and data from http://mplnet.gsfc.nasa.gov/ops.html.

MPL

Micro-Pulse Lidar



MPL digital image courtesy of Welch Mechanical Designs, LLC.

Figure 2: Representation of the Micro-pulse Lidar instrument. Image used from InstrumentReport212783_2005.pdf.

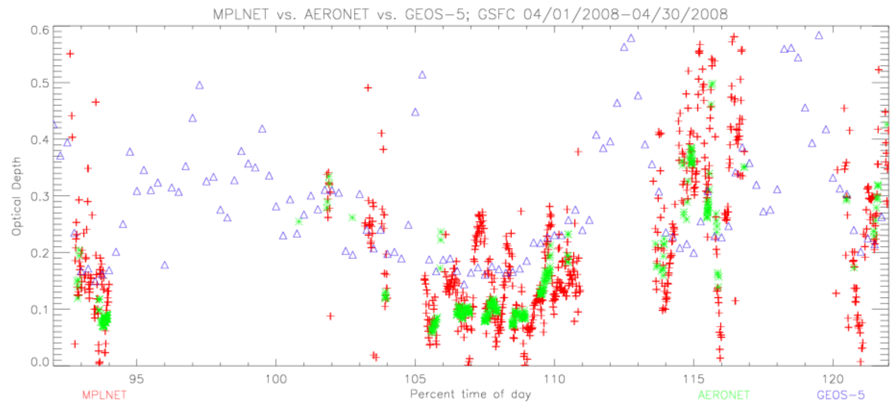


Figure 2: Comparison graph of the surface Aerosol Optical Depth for the MPLNET Gridded AOD (+), the AERONET coincident times AOD (*), and the GEOS-5 AOD (Triangle) from a Level 15a MPLNET file for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Ellsworth J. Welton and Peter Colarco.

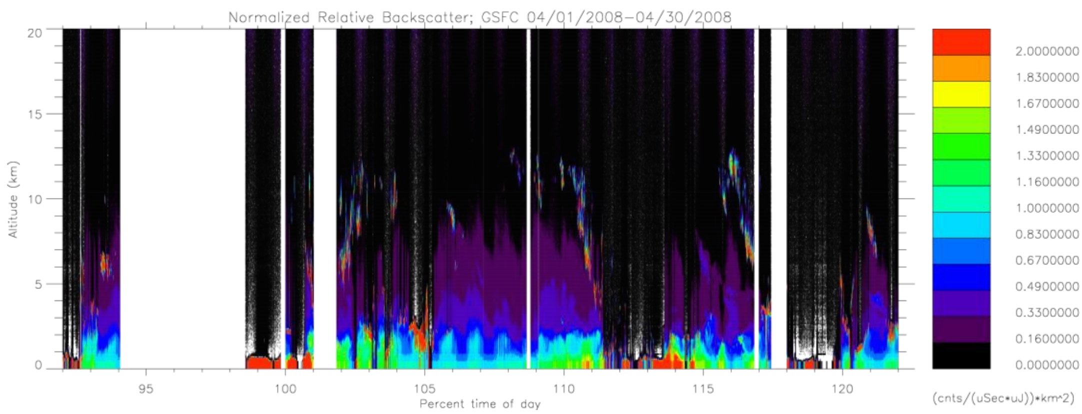


Figure 4: Graph of the Normalized Relative Backscatter variable from a Level 1 MPLNET file for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Ellsworth J. Welton.

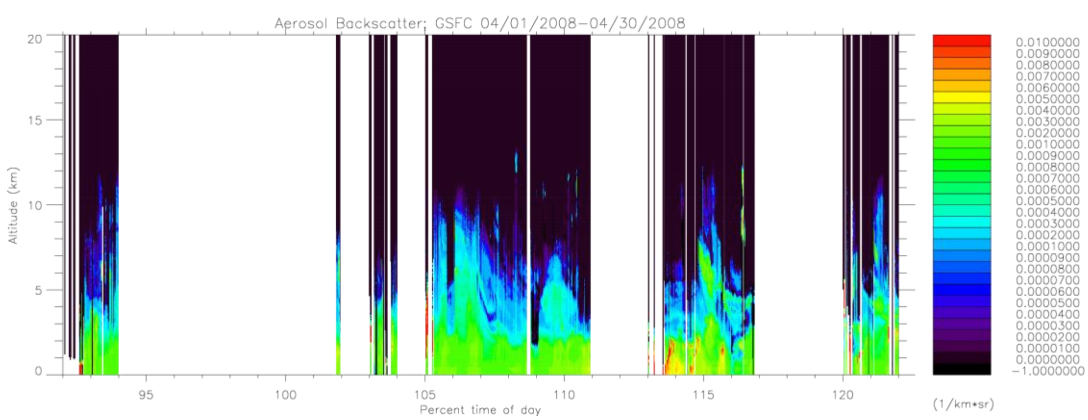


Figure 5: Graph of the Aerosol Backscatter variable from a Level 15a MPLNET file for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Ellsworth J. Welton.

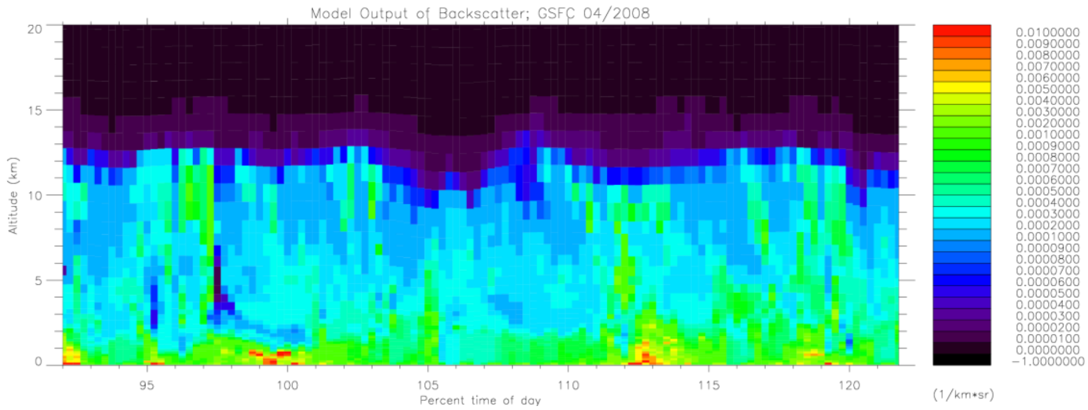


Figure 6: Graph of the Model representation of the Aerosol Backscatter variable for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Peter Colarco.

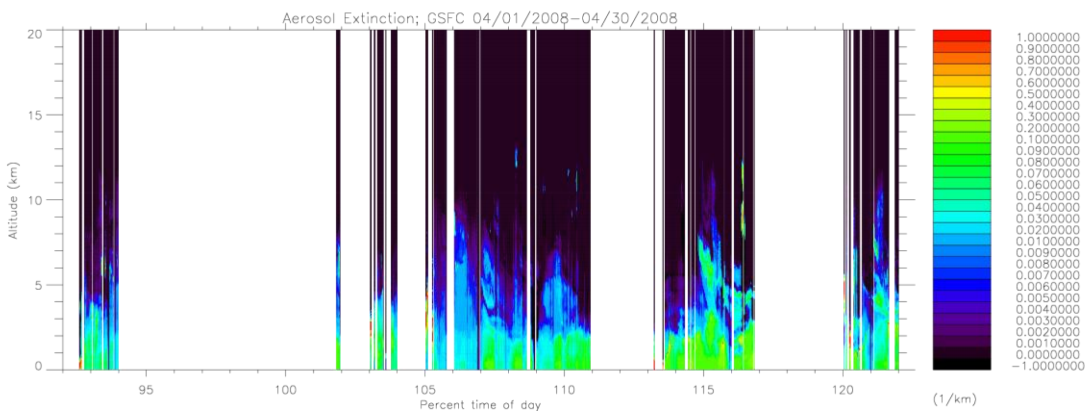


Figure 7: Graph of the Aerosol Extinction variable from a Level 15a MPLNET file for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Ellsworth J. Welton.

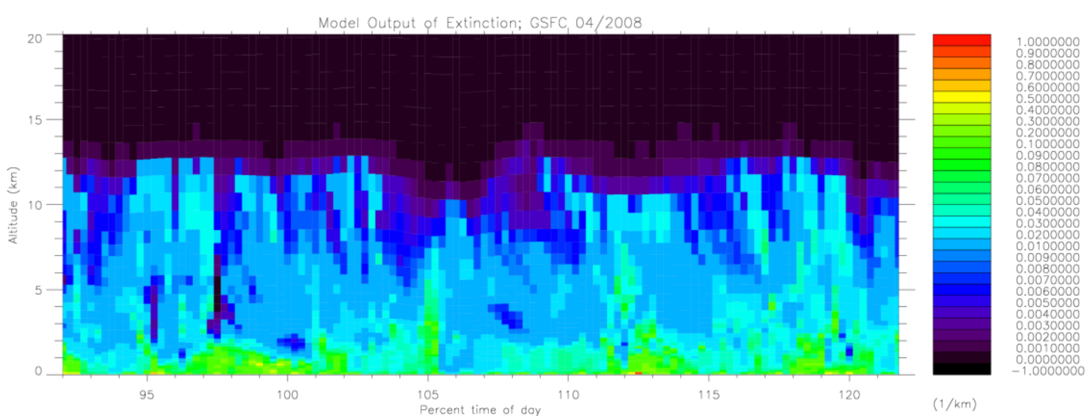


Figure 8: Graph of the Model representation of the Aerosol Extinction variable for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Peter Colarco.

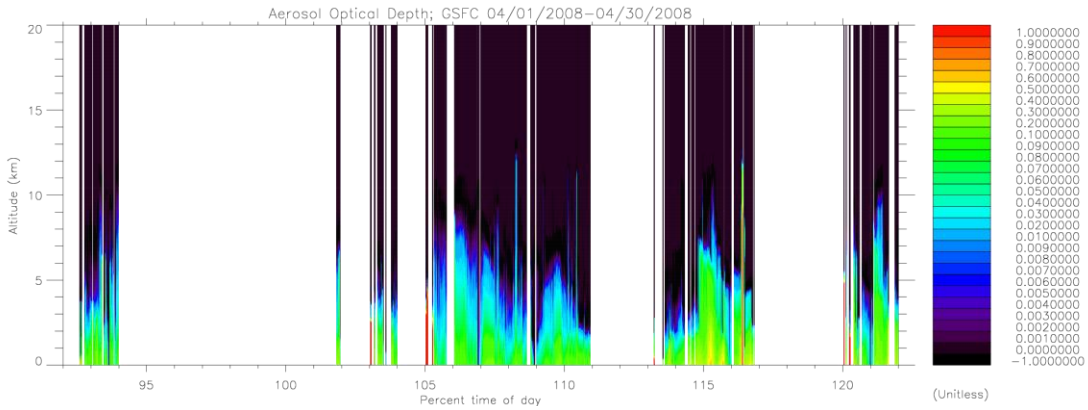


Figure 9: Graph of the Aerosol Optical Depth variable from a Level 15a MPLNET file for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Ellsworth J. Welton.

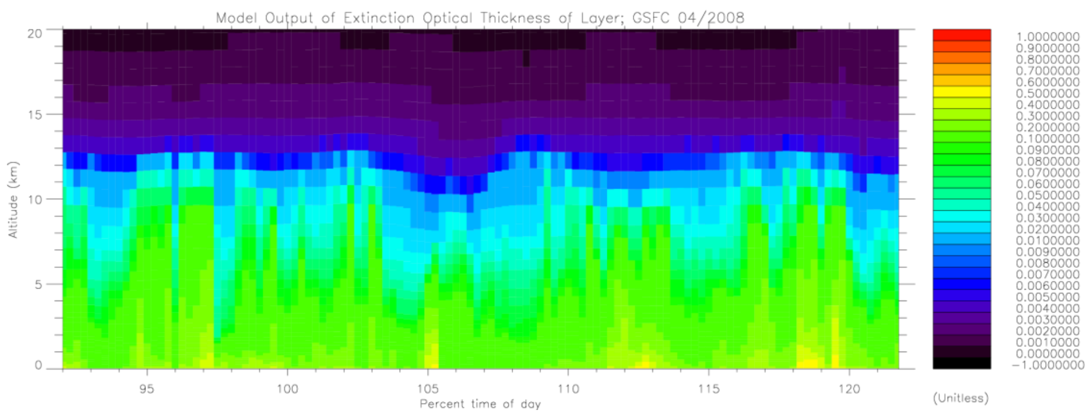


Figure 10: Graph of the Model representation of the Aerosol Optical Depth variable for the time span from April 1 to April 30, 2008. Plot created using IDL, and using data provided by Peter Colarco.

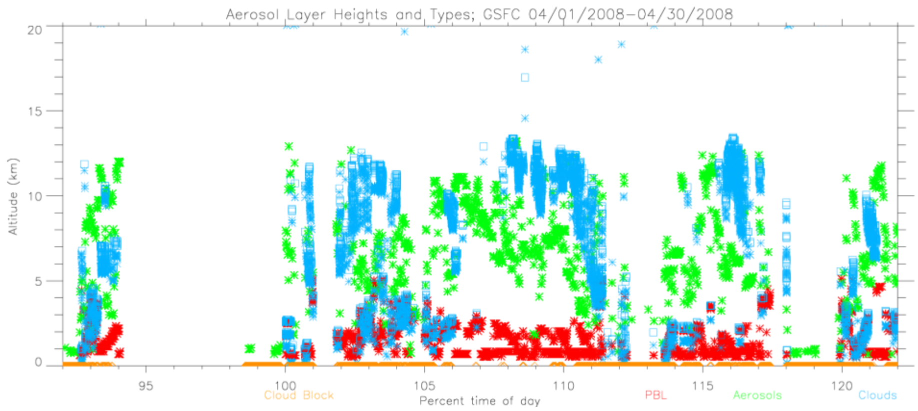


Figure 11: Graph of the various Aerosol layer heights and types from a Level 15b MPLNET file for the time span from April 1 to April 30, 2008. Cloud Block (diamond), PBL (red *), Aerosol Layer (green *), cloud base (blue *), and cloud top (square). Plot created using IDL, and using data provided by Ellsworth J. Welton.

Appendix A: *Read Me*

“rangeplot.pro” [Main Extended Plotting Program]

Signature= rangeplot, vFilename, varName

The ‘vFilename’ is the *full* filename of the .cdf file, and ‘varName’ is the variable code to be plotted.

Used to create plots of .cdf files. Takes a *user* specified .cdf file as input, and *user* specified variable name. The filename input into this program MUST be the full file path. This program uses the ‘varName’ variable to determine which level graph to plot. The user will be prompted for a date range to be plotted.

.pro files directly required to run include; *setday.pro, dayinc.pro, level1_lplot.pro, level15a_bs_lplot.pro, level15a_ex_lplot.pro, level15a_od_lplot.pro, level15b_lplot.pro*. In addition, you need all sub-files, defined below.

“setday.pro” [Procedure to set the Date part of the filename]

Signature= setday, vFile, vDate

The ‘vFile’ can either be the *full* path filename or the .cdf filename of the file to be changed, and ‘vDate’ is the date value you wish to set the file to. ‘vDate’ *MUST* be in the format of yyyymmdd!

This procedure is used to directly change the date value of a MPLNET file. This is done by just setting the value of the date part of the filename to the specified date.

There are no required .pro files to run this procedure.

“dayinc.pro” [Procedure to increment the Day by 1]

Signature= dayinc, vDate, vNewDate

The ‘vDate’ is the current date, and ‘vNewDate’ is the new date after the increase.

This procedure takes a date, which MUST be in yyyymmdd format, and converts it to a Julian day. This date is then incremented by one, and converted back into the original format and returned via the ‘vNewDate’ variable.

There are no required .pro files to run this procedure.

“level1_lplot.pro” [Procedure to plot the NRB from level 1 files]

Signature= level1_lplot, vFileArray, vRange

The ‘vFileArray’ is an array of *full* file names, and ‘vRange’ is a date range.

This procedure takes an array of *full* path file names, and a range of dates. If you would like to have one day, you must input the same date twice. For example, for 20080416, you must input 20080416-20080416. This procedure reads each file and retrieves the necessary variables. These variables are then placed into an expanding array. The plot is setup, and the values are plotted using a previous lib procedure already in place. This procedure saves the graph as a .ps file. The .ps files are located in “/aerutils/mplnet/output/plots/postscript/*filename.ps*”. The .ps file is saved with a descriptive filename for the variable graphed, and the designation ‘range_output’. For example, the completed graph of the backscatter variable is saved as, “bs_range_output.ps”.

.pro files directly required to run include; *level1_read.pro, datestring.pro, plotgrid.pro*, makekey.pro*.*

"level1_read.pro" [Procedure to read the level 1 files]

Signature= level1_read, vFilename, nrb_out, time_out, range_out

The 'vFilename' is the *full* file path name, and the '…_out' variables are the file variables.

This procedure reads in a file, and pulls out the variables from the file.

There are no required .pro files to run this procedure.

"datestring.pro" [Specific procedure to set a date string for plot]

Signature= datestring, vRange, date

The 'vRange' is the input date range, and the 'date' is the output of the new string formatted range.

This procedure is for setting up the formatted string for a range of dates to be placed in the plot graph title.

There are no required .pro files to run this procedure.

"level15a_bs_lplot.pro" [Procedure to plot the Bs from level 15a files]

Signature= level15a_bs_lplot, vFileArray, vRange

The 'vFileArray' is an array of *full* file names, and 'vRange' is a date range.

This procedure takes an array of *full* path file names, and a range of dates. If you would like to have one day, you must input the same date twice. For example, for 20080416, you must input 20080416-20080416. This procedure reads each file and retrieves the necessary variables. These variables are then placed into an expanding array. The plot is setup, and the values are plotted using a previous lib procedure already in place. This procedure saves the graph as a .ps file. The .ps files are located in "/aerutils/mplnet/output/plots/postscript/*filename.ps*". The .ps file is saved with a descriptive filename for the variable graphed, and the designation 'range_output'. For example, the completed graph of the backscatter variable is saved as, "bs_range_output.ps".

.pro files directly required to run include; *level15a_read.pro, datestring.pro, plotgrid.pro*, makekey.pro*.*

"level15a_read.pro" [Procedure to read the level 15a files]

Signature= level15a_read, vFilename, backscatter_out, extinction_out, optical_depth_out, time_out, range_out, aod_out, atime_out

The 'vFilename' is the *full* file path name, and the '…_out' variables are the file variables.

This procedure reads in a file, and pulls out the variables from the file.

There are no required .pro files to run this procedure.

"level15a_ex_lplot.pro" [Procedure to plot the Ex from level 15a files]

Signature= level15a_ex_lplot, vFileArray, vRange

The 'vFileArray' is an array of *full* file names, and 'vRange' is a date range.

This procedure takes an array of *full* path file names, and a range of dates. If you would like to have one day, you must input the same date twice. For example, for 20080416, you must input 20080416-20080416. This procedure reads each file and retrieves the necessary variables. These variables are then placed into an expanding array. The plot is setup, and the values are plotted using a previous lib procedure already in place. This procedure saves the graph as a .ps file. The .ps files are located in "/aerutils/mplnet/output/plots/postscript/*filename.ps*". The .ps file is saved with a descriptive filename for the variable graphed, and the designation 'range_output'. For example, the completed graph of the backscatter variable is saved as, "bs_range_output.ps".

.pro files directly required to run include; *level15a_read.pro, datestring.pro, plotgrid.pro*, makekey.pro**.

"level15a_od_lplot.pro" [Procedure to plot the Od from level 15a files]

Signature= level15a_od_lplot, vFileArray, vRange

The 'vFileArray' is an array of *full* file names, and 'vRange' is a date range.

This procedure takes an array of *full* path file names, and a range of dates. If you would like to have one day, you must input the same date twice. For example, for 20080416, you must input 20080416-20080416. This procedure reads each file and retrieves the necessary variables. These variables are then placed into an expanding array. The plot is setup, and the values are plotted using a previous lib procedure already in place. This procedure saves the graph as a .ps file. The .ps files are located in "/aerutils/mplnet/output/plots/postscript/*filename.ps*". The .ps file is saved with a descriptive filename for the variable graphed, and the designation 'range_output'. For example, the completed graph of the backscatter variable is saved as, "bs_range_output.ps".

.pro files directly required to run include; *level15a_read.pro, datestring.pro, plotgrid.pro*, makekey.pro**.

"level15b_lplot.pro" [Procedure to plot the layers from level 15b files]

Signature= level15b_lplot, vFileArray, vRange

The 'vFileArray' is an array of *full* file names, and 'vRange' is a date range.

This procedure takes an array of *full* path file names, and a range of dates. If you would like to have one day, you must input the same date twice. For example, for 20080416, you must input 20080416-20080416. This procedure reads each file and retrieves the necessary variables. These variables are then placed into an expanding array. The plot is setup, and the values are plotted using a previous lib procedure already in place. This procedure saves the graph as a .ps file. The .ps files are located in "/aerutils/mplnet/output/plots/postscript/*filename.ps*". The .ps file is saved with a descriptive filename for the variable graphed, and the designation 'range_output'. For example, the completed graph of the backscatter variable is saved as, "bs_range_output.ps".

.pro files directly required to run include; *level15b_read.pro, datestring.pro, plotgrid.pro*, makekey.pro**.

“level15b_read.pro” [Procedure to read the level 15b files]

Signature= level15b_read, vFilename, pbl_out, aerosol_out, cb_out, ct_out, beam_out, time_out, range_out

The ‘vFilename’ is the *full* file path name, and the ‘…_out’ variables are the file variables.

This procedure reads in a file, and pulls out the variables from the file.

There are no required .pro files to run this procedure.

“aod_plot.pro” [Procedure to compare the gridded-AOD to AOD]

Signature= aod_plot, vFilename, vVarFilename, vRange

The ‘vFilename’ is the *full* file path name, ‘vVarFilename’ is the file that contains the variable information (…inst…), and ‘vRange’ is a date range.

This procedure is for the purposes of comparing the differences of the MPLNET Gridded AOD, AERONET time AOD, and GEOS-5 AOD values. All files MUST be *full* file paths, and the date range MUST be in the form of yyyymmdd-yyyymmdd. The procedure plots all three variable’s AODs at the *surface* on one plot.

.pro files directly required to run include; *setday.pro, dayinc.pro, level15a_read.pro, read_lidartrack.pro*, datestring.pro*.

“modelplot.pro” [Main Model Plotting Procedure]

Signature= modelplot, vVarFilename, vPlotFilename, varName

The ‘vVarFilename’ is the file that contains the variable information (…inst…), ‘vPlotFilename’ is the file that contains the plotting information (…met…), and ‘varName’ is a *user* specified variable.

Used to create plots of .hdf files. Takes two *user* specified .hdf files as input, and *user* specified variable name. The filename inputs into this program MUST be the full file path. This program uses the ‘varName’ variable to determine which model graph to plot. This data is for a month, or determined by the file data.

.pro files directly required to run include; *modelplot_bs.pro, modelplot_ex.pro, modelplot_tau.pro*. In addition, you need all sub-files defined below.

“modelplot_bs.pro” [Procedure to plot the Bs model data]

Signature= modelplot_bs, vVarFilename, vPlotFilename

The ‘vVarFilename’ is the file that contains the variable information (…inst…), ‘vPlotFilename’ is the file that contains the plotting information (…met…).

This procedure takes the two files and plots them using duplicate plotting techniques as the level plots but using the data from the model. This procedure saves the graph as a .ps file. The .ps files are located in “/aerutils/mplnet/output/plots/postscript/*filename.ps*”. The .ps file is saved with a descriptive filename for the variable graphed, and the designation ‘model_output’. For example, the completed graph of the od variable is saved as, “tau_model_output.ps”.

.pro files directly required to run include; *read_lidartrack.pro*, read_lidartrack_met.pro**.

“modelplot_ex.pro” [Procedure to plot the Ex model data]

Signature= modelplot_ex, vVarFilename, vPlotFilename

The ‘vVarFilename’ is the file that contains the variable information (…inst…), ‘vPlotFilename’ is the file that contains the plotting information (…met…).

This procedure takes the two files and plots them using duplicate plotting techniques as the level plots but using the data from the model. This procedure saves the graph as a .ps file. The .ps files are located in “/aerutils/mplnet/output/plots/postscript/*filename.ps*”. The .ps file is saved with a descriptive filename for the variable graphed, and the designation ‘model_output’. For example, the completed graph of the od variable is saved as, “tau_model_output.ps”.

.pro files directly required to run include; *read_lidartrack.pro*, read_lidartrack_met.pro**.

“modelplot_tau.pro” [Procedure to plot the Tau model data]

Signature= modelplot_tau, vVarFilename, vPlotFilename

The ‘vVarFilename’ is the file that contains the variable information (…inst…), ‘vPlotFilename’ is the file that contains the plotting information (…met…).

This procedure takes the two files and plots them using duplicate plotting techniques as the level plots but using the data from the model. This procedure saves the graph as a .ps file. The .ps files are located in “/aerutils/mplnet/output/plots/postscript/*filename.ps*”. The .ps file is saved with a descriptive filename for the variable graphed, and the designation ‘model_output’. For example, the completed graph of the od variable is saved as, “tau_model_output.ps”.

.pro files directly required to run include; *read_lidartrack.pro*, read_lidartrack_met.pro**.

“sitemap.pro” [Procedure to create a map]

Signature= .run sitemap.pro

Procedure creates a map of the MPLNET site locations.

There are no required .pro files to run this procedure.

“daydec.pro” [Procedure to increment the Day by 1]

Signature= daydec, vDate, vNewDate

The ‘vDate’ is the current date, and ‘vNewDate’ is the new date after the increase.

This procedure takes a date, which MUST be in yyyymmdd format, and converts it to a Julian day. This date is then decremented by one, and converted back into the original format and returned via the ‘vNewDate’ variable.

There are no required .pro files to run this procedure.

“layers_oplot.pro” [Incomplete Procedure to Overlay Layers on NRB Graph]**

Signature= vFileL1, vFileL2, vRange

The ‘vFileL1’ is an array of Level 1 *full* file names, ‘vFileL2’ is an array of Level 15b *full* file names, and ‘vRange’ is a date range.

This procedure takes an array of *full* path file names, and a range of dates. If you would like to have one day, you must input the same date twice. For example, for 20080416, you must input 20080416-20080416. This procedure reads each file and retrieves the necessary variables. These variables are then placed into an expanding array. The plot is setup, and the values are plotted using a previous lib procedure already in place. This procedure saves the graph as a .ps file. The .ps files are located in "/aerutils/mplnet/output/plots/postscript/*filename.ps*". The .ps file is saved with a descriptive filename for the variable graphed, and the designation 'overplot_output'. For example, the completed graph of the NRB/Layers over plot is saved as, "layer_overplot_output.ps".

.pro files directly required to run include; *setday.pro, dayinc.pro, level1_read.pro, level15b_read.pro, datestring.pro, plotgrid.pro*, makekey.pro**.

(* = Procedure already in place in IDL Library folder)