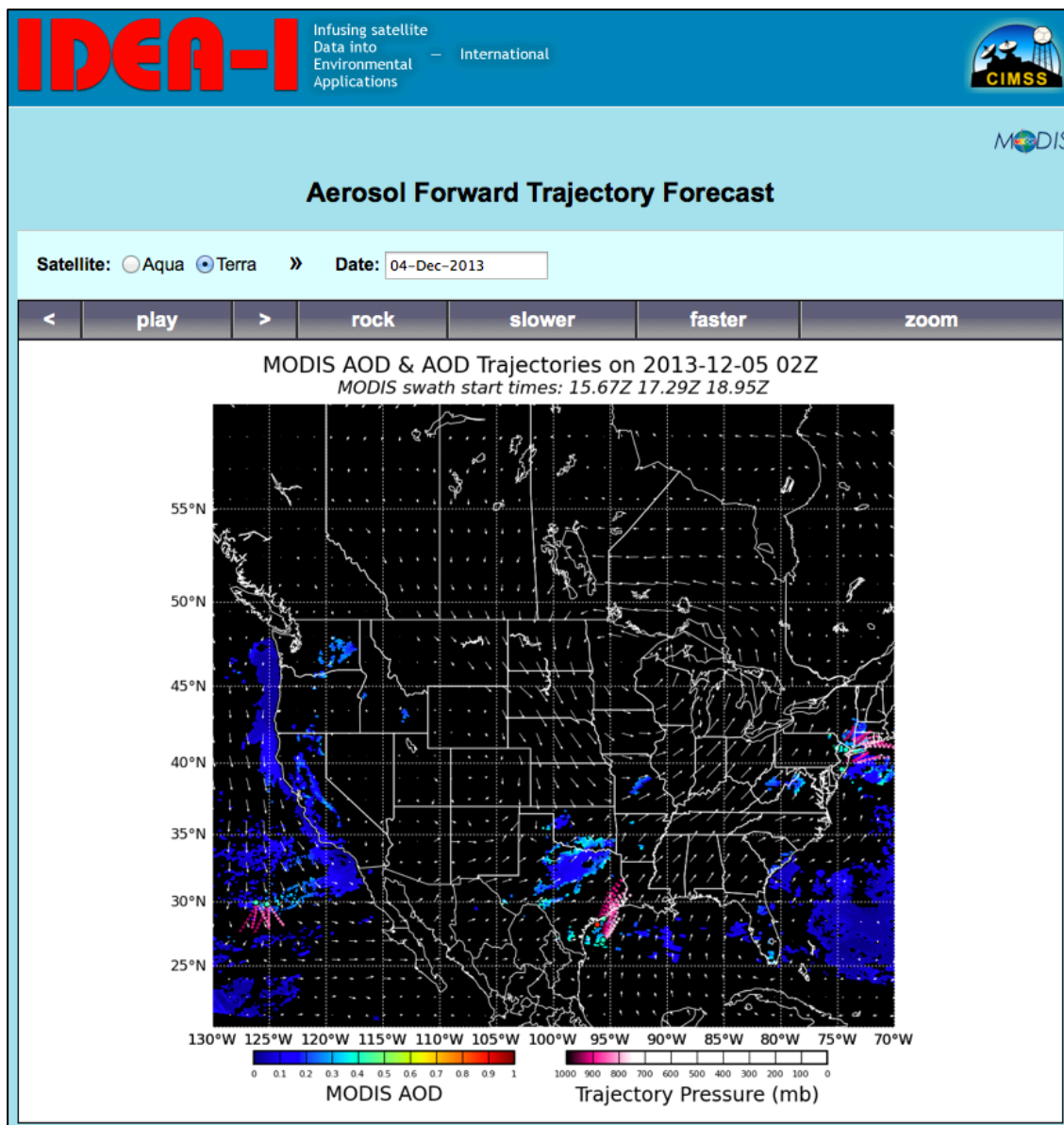# Installation Instructions for the International MODIS/AIRS Processing Package (IMAPP) Aerosol Air Quality Forecasting Application
IDEA-I: Infusing satellite Data into Environmental Applications – International

Version 1.1

University of Wisconsin-Madison, Space Science and Engineering Center (SSEC)
Supported by the NASA Science of Aqua and Terra Project

Beta Version – 13 December 2013

Table of Contents

Section 1: Introduction

1.1 Overview

This document contains instructions for installation and operation of the IMAPP MODIS Aerosol Air Quality Forecast and Analysis software package, IDEA-I, in support of direct broadcast users. It is a globally configurable package that:

- **Creates 48 hour 3 dimensional trajectory forecasts of aerosol pollution.**
  - **Terra and Aqua Moderate Resolution Imaging Spectroradiometer (MODIS)** MOD04 Aerosol Optical Depth (AOD) retrievals are used to identify regions of high aerosol loading (>0.4), and then a trajectory model is run to forecast the horizontal and vertical movement of the aerosols over the next 48 hours.

The output products consist of both a set of netCDF data files as well as hourly trajectory forecast images, which can be viewed as animations from a web browser using the PHP software that is included as part of this package.

The Aerosol IDEA-I Version 1.1 Beta software is distributed through the IMAPP ftp site:

ftp://ftp.ssec.wisc.edu/pub/IMAPP/beta/idea_aerosol/

Software, test data, and documentation may be downloaded from this site.

The trajectories are predicted using the NASA Langley trajectory code [Pierce and Fairlie, 1993]. More detailed information about the model is provided in the Appendix.

1.2 What's new in version 1.1?

- The trajectory model performance has been dramatically improved. The execution speed depends on the number of trajectories that are initialized. The performance should be 2-3x faster than for a similar number of trajectories process in version 1.0.
- A new main run script, IDEA-I.bash has been added.
- Many changes have been made to the configuration file, including the capability to point to different sources of GFS ancillary data for archived data processing.
- the PHP software, that allows display of the output image files, now allows the user to display more than one product per website.

1.3 System requirements

System requirements for the IDEA-I aerosol software are as follows:
- Intel or AMD CPU with 64-bit instruction support,
- 1 GB RAM,
- CentOS 5 64-bit Linux (or other compatible 64-bit Linux distribution),
- 2 GB disk space,
- Reliable and efficient internet connection for downloading the required ancillary data (approximately 400 MB per day).

Linux terminal commands included in these instructions assume the bash shell is used.

1.4 Input Data Requirements

1.4.1 MODIS

MODIS Collection 5 Aerosol Product HDF4 files (MOD04), and the MODIS Cloud Top Properties files (MOD06) are required to run the aerosol trajectory forecast. The files can either be created through the IMAPP MODIS Level-2 package, (http://cimss.ssec.wisc.edu/imapp/modis_level2_v2.2.shtml), or downloaded from the MODIS LAADS (http://ladsweb.nascom.nasa.gov/) archive or the LANCE near-real time data server (http://lance-modis.eosdis.nasa.gov/). The user can set the naming convention (either IMAPP or standard NASA filenames) in the aerosol configuration file.

1.5 Disclaimer

Original software scripts and automation included as part of this package are distributed under the GNU GENERAL PUBLIC LICENSE agreement version 3. Software included as part of this software package are copyrighted and licensed by their respective organizations, and distributed consistent with their licensing terms.

The University of Wisconsin-Madison Space Science and Engineering Center (SSEC) makes no warranty of any kind with regard to the IMAPP IDEA-I software or any accompanying documentation, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. SSEC does not indemnify any infringement of copyright, patent, or trademark through the use or modification of this software.

There is no expressed or implied warranty made to anyone as to the suitability of this software for any purpose. All risk of use is assumed by the user. Users agree not to hold SSEC, the University of Wisconsin-Madison, or any of its employees or assigns liable for any consequences resulting from the use of the IMAPP IDEA-I software.

Section 2: Installation and Configuration

2.1 Overview

This software will generate air quality forecast products from input Aqua and Terra MODIS data. It relies on setting up configuration files that point to the location of the input satellite and ancillary data files. Execution can be automated by using the cron unix scheduler.

2.2 Installation of IDEA-I Software

Download the following files from the IMAPP Beta ftp site:
ftp://ftp.ssec.wisc.edu/pub/IMAPP/beta/idea_aerosol/

`IMAPP_IDEA-I_Aerosol_V1.1_BETA.tar.gz`

Next, unpack the tarfile (a new directory named `IDEA-I_aerosol` will be created):

`tar xf IMAPP_IDEA-I_Aerosol_V1.1_BETA.tar.gz`

2.3 Setting Up Configuration Files

Edit the MODIS_aerosol.cfg configuration file located in the `IDEA-I_aerosol/config` directory of your installation. We suggest that you copy the configuration file into a separate file, like my_aerosol.cfg, and edit that file to use during execution. There are many local adjustments that can be made to your IDEA-I installation by setting variables in the configuration files. For instance, you can choose to download the required satellite data input files from a remote location, such as from the public University of Wisconsin ftp site (default), and you can choose how many days of data to keep on disk. Some important parameters are listed below.

Required parameters:
> OUTPUT_DIR =   Output Directory name
> REGION_NAME =   Domain name written into output netCDF file attribute
> TOP_LEFT_LON_LAT =   IDEA-I domain upper left lon/lat (ex: -130, 60)
> BOTTOM_RIGHT_LON_LAT = IDEA-I domain lower right lon/lat (ex: -70, 20)

MODIS Aqua and Terra Aerosol Trajectory specific parameters (Aqua example):
> A04TEMPLATE = NASA or IMAPP MODIS AOD file naming convention template
> > IMAPP Aqua Example: IMAPP - a1(.%y%j.%H%M.mod04.hdf
> > NASA Aqua Example: MYD04_L2.A%Y%j.%H%M.*.hdf
> A04LOCAL =   Local directory where MODIS Aerosol Optical Depth files are located
> A06TEMPLATE = NASA or IMAPP MODIS Cloud Top Properties file naming
> > convention template
> > IMAPP Aqua Example: IMAPP - a1(.%y%j.%H%M.mod06ct.hdf
> > NASA Aqua Example: MYD06_L2.A%Y%j.%H%M.*.hdf
> A06LOCAL = Local directory where MODIS Cloud Top Properties files are located

Other options allow users to download satellite data from a local server, and to fetch ancillary analysis (as opposed to forecast) data from the National Climatic Data Center (NCDC) when a

user wishes to process satellite data that is more than 2 days old. Please see the MODIS_aerosol.cfg file for a complete list of options.

2.4 Ancillary Data

National Centers for Environmental Prediction (NCEP) Global Forecast System (GFS) grib2 files are required as input to the trajectory forecast model. The required files are identified and downloaded as part of the IDEA-I execution. The site that the files are downloaded from can be set using the configuration files; the default is to fetch the files from the University of Wisconsin ancillary data site:

> ftp://ftp.ssec.wisc.edu/pub/eosdb/ancillary.

GFS grib2 formatted files from 00 hour to 60 hour forecasts at 6 hourly intervals are required. If your satellite data collection for a given instrument includes observations prior to 12 UTC for a given IDEA-I.bash execution, you will also need to download forecast fields from the previous day. Each of the grib2 files is about 35MB each; therefore, each day will require a download of about 400 MB. Both the MODIS aerosol and hyperspectral sounder ozone trajectory forecasts require the same ancillary files; if you are using both software packages, this download is required only once per day.

For dates and times that you wish to process that are more than a couple days old, then you should use analysis fields from the Global Data Assimilation System (GDAS) instead of forecast fields. These are distributed by the National Climatic Data Center (NCDC); the configuration files have entries that point to these files if you choose to use them.

Section 3: Creating MODIS Aerosol Trajectories

This section describes how to run MODIS IDEA-I aerosol to create trajectories of points identified as having high aerosol optical depths, defined as the MOD04 Optical_Depth_Land_And_Ocean product values above 0.4. A 48 hour trajectory model is run to forecast the movement of the aerosols in 3 dimensions over the chosen domain.

3.1 How can I get started quickly?

This section provides the basics for running a standard installation of IDEA-I for the creation of routine aerosol trajectory products. More detailed information can be found in the following sections.

1). Unpack the software as described in Section 2.2 and set the required parameters to match your domain and system in your configuration file (Section 2.3).
2). Set the IDEA_HOME variable to be your top level installation directory, for instance:

```
export IDEA_HOME=${HOME}/IDEA-I_aerosol
```

3). Test the software on your MODIS data using the –v (verbose) option using your own data from yesterday:

```
${IDEA_HOME}/IDEA-I.bash MODIS aerosol -vGXM Aqua- -d YESTERDAY -f
my_aerosol.cfg
```

As a test, running for YESTERDAY is safer than running for TODAY, since the source data for
TODAY may not yet be available; remember also that TODAY is defined in UTC time.

4). Add a daily run to crontab, using –d=TODAY, with scheduled executions times set to after all
daily source data is available.

15 20 * * * /home/IDEA-I_aerosol/IDEA-I.bash MODIS aerosol -v -X -G -M Terra- -d TODAY -f
my_aerosol.cfg

This entry will process the Terra MODIS aerosol and cloud product files from passes acquired
today.

15 22 * * * /home/IDEA-I_aerosol/IDEA-I.bash MODIS aerosol -v -X -G -F -M Aqua+ -d TODAY -f
my_aerosol.cfg > /home/IDEA-I_aerosol/logs/ideai_aqua_$(date +\%Y\%m\%d_\%H\%M).log
2>&1

This entry will process all Aqua MODIS aerosol and cloud product files from today, and will
include a download of the direct broadcast product files staged at the Wisconsin Direct Broadcast
direct broadcast ftp site (-F option). If your domain falls within the Contiguous US, you can use
these files as input with this option. If you use this option, you must ensure the FETCH options in
the config file are set to YES. It sends the output ascii text information to a date stamped log file
in the form of:

       ideai_aqua_YYYYMMDD_HHMM.log (eg: ideai_aqua_20130530_2257.log)

If your domain is wholly covered by Terra MODIS data from one UTC day, but the data is not all
available until the next UTC day, run with -d=YESTERDAY

5). If you want to display your output files or images, follow the instructions in Section 4.

3.1 MODIS IDEA-I aerosol driver script

The principal driver script for MODIS aerosol trajectories is
IDEA-I_aerosol/IDEA-I.bash, which wraps the perl processing script,
IDEA-I-MODISaerosol.pl, located in the IDEA-I_aerosol/bin directory. The script requires two
inputs, the sensor name and the product you wish to create:

Usage: ./IDEA-I.bash <SENSOR> <product> <SENSORproduct-specific-arguments>

       Valid <SENSOR> <product> combinations are:
              MODIS aerosol

It operates by default in a temporary directory within IDEA-I_aerosol/tmp. This location can be set
in the configuration file (TMPDIR). It creates intermediate files, including binary ancillary files
(converted from grib2) as well as final trajectory NetCDF (.nc) and image files (.png) and places
them in a date stamped output file directory as defined in the configuration file that you are using.
The default output file prefix is IDEA-I_aerosol/products/myRegion. The software requires the
following inputs, whose naming conventions, remote and local locations are set through the
configuration files:

a) MODIS collect 5 aerosol optical depth files (MOD04) files in either IMAPP or NASA naming conventions (t1*.mod04.hdf or MOD04_L2*.hdf for Terra or a1*.mod04.hdf or MYD04_L2*.hdf for Aqua). The direct broadcast IMAPP MODIS Level 2 package (http://cimss.ssec.wisc.edu/imapp/modis_level2_v2.2.shtml) includes the software to create these products from input MODIS Level 1B HDF4 files.

b) MODIS cloud top properties product files in either IMAPP or NASA naming conventions (t1*.mod06ct.hdf or MOD06_L2*.hdf for Terra or a1*.mod06ct.hdf or MYD06_L2*.hdf for Aqua). The direct broadcast IMAPP MODIS Level 2 package (http://cimss.ssec.wisc.edu/imapp/modis_level2_v2.2.shtml) includes the software to create these products from input MODIS Level 1B HDF4 files.

c) National Centers for Environmental Prediction (NCEP) Global Forecast System (GFS) grib2 0 to 60 hour forecasts files. The required files are automatically identified and downloaded from the remote location identified in the configuration file.

To execute the IDEA-I processing script for MODIS aerosols, first set the top level IDEA-I directory, for example:

```
export IDEA_HOME=${HOME}/IDEA-I_aerosol
```

Execute the run script using –h to print out a complete list of options. A subset of options are shown:

```
$IDEA_HOME/IDEA-I.bash MODIS aerosol -h

  -v  run in verbose mode. Default is quiet.
  -f  configuration file – no default.
  -G  update GFS data from remote FTP site. Default is don't update at runtime
  -F  Allow fetching of Satellite Data remotely (Must also set FETCH option in
        config file to YES).  Default is don't allow satellite data fetching.
  -X  execute trajectory model. Default is don't run.
  -M  Satellite name and data time. No Defaults. Options are:
      Aqua+  Processing Aqua data hdf files.  Indicates all data observations
      occur after 12 UTC for the given day.  This lets the IDEA-I software know
      how much ancillary data to gather, one day or two.
      Aqua-  Processing Aqua data hdf files.  Indicates at least some data
      observations occur before 12 UTC for the given day.

      Terra+  Processing Aqua data hdf files.  Indicates all data
      observations occur after 12 UTC for the given day.  This lets the IDEA-I
      software know how much ancillary data to gather, one day or two
      Terra-  Processing Aqua data hdf files.  Indicates at least some data
      observations occur before 12 UTC for the given day.
  -d  Date – Can use either YYYYMMDD, TODAY, YESTERDAY
```

For example, executing the script with this set of options:

```
${IDEA_HOME}/IDEA-I.bash MODIS aerosol -v -G -X -M Aqua+ -d YESTERDAY -f
my_aerosol.cfg

                                    or

${IDEA_HOME}/IDEA-I.bash MODIS aerosol -vGXM Aqua+ -d YESTERDAY -f
my_aerosol.cfg
```

will create a 48 hour aerosol trajectory forecast based upon the Aqua MODIS data from yesterday, which only included data collected after 12 UTC, covering the domain defined in ${IDEA_HOME}/config/my_aerosol.cfg. It will print a stream of text (verbose mode) as it progresses, and will update the NWP (GFS) data sets as part of this execution. The output products will be placed in the default location defined in my configuration file ({IDEA-I}/products/myRegion) with the extension that includes MODISaerosol/<Aqua or Terra>/Date (YYYYMMDD).

```
${IDEA_HOME}/IDEA-I.bash MODIS aerosol -vGXM Terra- -d 20130714 –f
Terra_aerosol.cfg
```

This command will update the ancillary GFS files required to create a 48 hour aerosol trajectory forecast based upon the Terra MODIS data from 14 July 2013, which included at least some data observations prior to 12 UTC on that day. This means that NWP grids for both 14 July and 13 July 2013 are updated as part of this execution. The configuration file (Terra_aerosol.cfg) must have only the GFS analysis section for archive processing uncommented, the GFSFETCH=YES parameter set and the other two GFS sections commented out

If I then execute this command:

```
${IDEA_HOME}/IDEA-I.bash MODIS aerosol –vXM Terra- -d 20130714 –f
Terra_aerosol.cfg
```

the trajectory model will be executed. You may wish to separate the execution of the trajectory model and the ancillary file downloads, as it requires about 400 MB of GFS grib2 files for each day's execution.

3.2 MODIS aerosol IDEA-I output files

A successful execution of the IDEA-I MODIS aerosol trajectory forecast will result in a text log file containing details about the trajectory forecast model run (GFS_traject_3d_v01.log) as well as NetCDF data and .png image files covering your domain placed in the output file directory.

3.2.1 Output NetCDF Data files

Two data files in NetCDF format are produced for each IDEA-I aerosol run.

One contains both native resolution MODIS aerosol optical depth pass values as well as aggregated gridded aerosol and GFS model grid data. The naming convention for these files is:

 MODISaerosol(X)_grid_48hr_YYYYMMDD.nc (ex: **MODISaerosolA_grid_48hr_20131110.nc**)

Where X is either A for Aqua or T for Terra. These files contain separate data arrays for each pass acquired for that day in both native format resolution (10 km) as well as aggregated gridded format resolution (default tis 3-10km pixels by 4-10km pixels – set using the SUPIX and NPIX variables in your configuration file). The GFS fields are also included that cover your domain for the 48 hour time step. These fields are dimensioned "west_east" and "north_south" and are presented in 11 discrete time steps, from model initialization until 60 hours later, in 6 hour steps. The first available MODIS AOD data may be several hours after the trajectory model initialization time, depending upon your earth location and on the MODIS mission (Terra or Aqua) that you choose to use. Thus the 48 hour duration of the trajectory forecast is typical, but not precise.

A header file dump of a typical output NetCDF gridded data file consisting of 3 passes for the given day is shown below:

```
netcdf MODISaerosolT_grid_48hr_20131110 {
dimensions:
        time = UNLIMITED ; // (11 currently)
        west_east = 121 ;
        north_south = 81 ;
        across_track = 135 ;
        along_track_001 = 252 ;
        along_track_002 = 450 ;
        along_track_003 = 342 ;
        across_aggr = 45 ;
        along_aggr_001 = 84 ;
        along_aggr_002 = 150 ;
        along_aggr_003 = 114 ;
variables:
        float Aggr_AOD_001(along_aggr_001, across_aggr) ;
                Aggr_AOD_001:long_name = "aggregated pixel AOD swath 001" ;
                Aggr_AOD_001:_FillValue = -999.f ;
                Aggr_AOD_001:units = "dimensionless" ;
                Aggr_AOD_001:coordinates = "LonAggr_001 LatAggr_001" ;
        float Aggr_AOD_002(along_aggr_002, across_aggr) ;
                Aggr_AOD_002:long_name = "aggregated pixel AOD swath 002" ;
                Aggr_AOD_002:_FillValue = -999.f ;
                Aggr_AOD_002:units = "dimensionless" ;
                Aggr_AOD_002:coordinates = "LonAggr_002 LatAggr_002" ;
        float Aggr_AOD_003(along_aggr_003, across_aggr) ;
                Aggr_AOD_003:long_name = "aggregated pixel AOD swath 003" ;
                Aggr_AOD_003:_FillValue = -999.f ;
                Aggr_AOD_003:units = "dimensionless" ;
                Aggr_AOD_003:coordinates = "LonAggr_003 LatAggr_003" ;
        int Aggr_Counter_001(along_aggr_001, across_aggr) ;
                Aggr_Counter_001:long_name = "Valid MODIS pixels per aggregated pixel for swath 001" ;
                Aggr_Counter_001:_FillValue = -999 ;
                Aggr_Counter_001:units = "dimensionless" ;
                Aggr_Counter_001:coordinates = "LonAggr_001 LatAggr_001" ;
        int Aggr_Counter_002(along_aggr_002, across_aggr) ;
                Aggr_Counter_002:long_name = "Valid MODIS pixels per aggregated pixel for swath 002" ;
                Aggr_Counter_002:_FillValue = -999 ;
                Aggr_Counter_002:units = "dimensionless" ;
                Aggr_Counter_002:coordinates = "LonAggr_002 LatAggr_002" ;
        int Aggr_Counter_003(along_aggr_003, across_aggr) ;
                Aggr_Counter_003:long_name = "Valid MODIS pixels per aggregated pixel for swath 003" ;
                Aggr_Counter_003:_FillValue = -999 ;
                Aggr_Counter_003:units = "dimensionless" ;
                Aggr_Counter_003:coordinates = "LonAggr_003 LatAggr_003" ;
        int Aggr_inROI_001(along_aggr_001, across_aggr) ;
                Aggr_inROI_001:long_name = "aggregated pixel in region of interest for swath 001" ;
                Aggr_inROI_001:_FillValue = -999 ;
                Aggr_inROI_001:units = "dimensionless" ;
                Aggr_inROI_001:coordinates = "LonAggr_001 LatAggr_001" ;
        int Aggr_inROI_002(along_aggr_002, across_aggr) ;
                Aggr_inROI_002:long_name = "aggregated pixel in region of interest for swath 002" ;
                Aggr_inROI_002:_FillValue = -999 ;
                Aggr_inROI_002:units = "dimensionless" ;
                Aggr_inROI_002:coordinates = "LonAggr_002 LatAggr_002" ;
        int Aggr_inROI_003(along_aggr_003, across_aggr) ;
                Aggr_inROI_003:long_name = "aggregated pixel in region of interest for swath 003" ;
                Aggr_inROI_003:_FillValue = -999 ;
                Aggr_inROI_003:units = "dimensionless" ;
                Aggr_inROI_003:coordinates = "LonAggr_003 LatAggr_003" ;
        float Aggr_psfc_001(along_aggr_001, across_aggr) ;
                Aggr_psfc_001:long_name = "aggregation psfc time-interpolated for swath 001" ;
```

```
        Aggr_psfc_001:_FillValue = -999.f ;
        Aggr_psfc_001:units = "hPa" ;
        Aggr_psfc_001:coordinates = "LonAggr_001 LatAggr_001" ;
float Aggr_psfc_002(along_aggr_002, across_aggr) ;
        Aggr_psfc_002:long_name = "aggregation psfc time-interpolated for swath 002" ;
        Aggr_psfc_002:_FillValue = -999.f ;
        Aggr_psfc_002:units = "hPa" ;
        Aggr_psfc_002:coordinates = "LonAggr_002 LatAggr_002" ;
float Aggr_psfc_003(along_aggr_003, across_aggr) ;
        Aggr_psfc_003:long_name = "aggregation psfc time-interpolated for swath 003" ;
        Aggr_psfc_003:_FillValue = -999.f ;
        Aggr_psfc_003:units = "hPa" ;
        Aggr_psfc_003:coordinates = "LonAggr_003 LatAggr_003" ;
float Cloud_Effective_Emissivity_001(along_track_001, across_track) ;
        Cloud_Effective_Emissivity_001:long_name = "cloud effective emissivity for swath number 001" ;
        Cloud_Effective_Emissivity_001:_FillValue = -999.f ;
        Cloud_Effective_Emissivity_001:units = "dimensionless" ;
        Cloud_Effective_Emissivity_001:coordinates = "Lon_001 Lat_001" ;
float Cloud_Effective_Emissivity_002(along_track_002, across_track) ;
        Cloud_Effective_Emissivity_002:long_name = "cloud effective emissivity for swath number 002" ;
        Cloud_Effective_Emissivity_002:_FillValue = -999.f ;
        Cloud_Effective_Emissivity_002:units = "dimensionless" ;
        Cloud_Effective_Emissivity_002:coordinates = "Lon_002 Lat_002" ;
float Cloud_Effective_Emissivity_003(along_track_003, across_track) ;
        Cloud_Effective_Emissivity_003:long_name = "cloud effective emissivity for swath number 003" ;
        Cloud_Effective_Emissivity_003:_FillValue = -999.f ;
        Cloud_Effective_Emissivity_003:units = "dimensionless" ;
        Cloud_Effective_Emissivity_003:coordinates = "Lon_003 Lat_003" ;
float LatAggr_001(along_aggr_001, across_aggr) ;
        LatAggr_001:long_name = "latitude for aggregated pixel for swath 001" ;
        LatAggr_001:_FillValue = -999.f ;
        LatAggr_001:units = "degrees_north" ;
        LatAggr_001:valid_max = 90.f ;
        LatAggr_001:valid_min = -90.f ;
float LatAggr_002(along_aggr_002, across_aggr) ;
        LatAggr_002:long_name = "latitude for aggregated pixel for swath 002" ;
        LatAggr_002:_FillValue = -999.f ;
        LatAggr_002:units = "degrees_north" ;
        LatAggr_002:valid_max = 90.f ;
        LatAggr_002:valid_min = -90.f ;
float LatAggr_003(along_aggr_003, across_aggr) ;
        LatAggr_003:long_name = "latitude for aggregated pixel for swath 003" ;
        LatAggr_003:_FillValue = -999.f ;
        LatAggr_003:units = "degrees_north" ;
        LatAggr_003:valid_max = 90.f ;
        LatAggr_003:valid_min = -90.f ;
float Lat_001(along_track_001, across_track) ;
        Lat_001:long_name = "latitude for swath number 001" ;
        Lat_001:_FillValue = -999.f ;
        Lat_001:units = "degrees_north" ;
        Lat_001:valid_max = 90.f ;
        Lat_001:valid_min = -90.f ;
float Lat_002(along_track_002, across_track) ;
        Lat_002:long_name = "latitude for swath number 002" ;
        Lat_002:_FillValue = -999.f ;
        Lat_002:units = "degrees_north" ;
        Lat_002:valid_max = 90.f ;
        Lat_002:valid_min = -90.f ;
float Lat_003(along_track_003, across_track) ;
        Lat_003:long_name = "latitude for swath number 003" ;
        Lat_003:_FillValue = -999.f ;
        Lat_003:units = "degrees_north" ;
        Lat_003:valid_max = 90.f ;
        Lat_003:valid_min = -90.f ;
float Latitude(north_south) ;
```

```
        Latitude:long_name = "latitude" ;
        Latitude:_FillValue = -999.f ;
        Latitude:units = "degrees_north" ;
        Latitude:valid_max = 90.f ;
        Latitude:valid_min = -90.f ;
float LonAggr_001(along_aggr_001, across_aggr) ;
        LonAggr_001:long_name = "longitude for aggregated pixel for swath 001" ;
        LonAggr_001:_FillValue = -999.f ;
        LonAggr_001:units = "degrees_east" ;
        LonAggr_001:valid_max = 180.f ;
        LonAggr_001:valid_min = -180.f ;
float LonAggr_002(along_aggr_002, across_aggr) ;
        LonAggr_002:long_name = "longitude for aggregated pixel for swath 002" ;
        LonAggr_002:_FillValue = -999.f ;
        LonAggr_002:units = "degrees_east" ;
        LonAggr_002:valid_max = 180.f ;
        LonAggr_002:valid_min = -180.f ;
float LonAggr_003(along_aggr_003, across_aggr) ;
        LonAggr_003:long_name = "longitude for aggregated pixel for swath 003" ;
        LonAggr_003:_FillValue = -999.f ;
        LonAggr_003:units = "degrees_east" ;
        LonAggr_003:valid_max = 180.f ;
        LonAggr_003:valid_min = -180.f ;
float Lon_001(along_track_001, across_track) ;
        Lon_001:long_name = "longitude for swath number 001" ;
        Lon_001:_FillValue = -999.f ;
        Lon_001:units = "degrees_east" ;
        Lon_001:valid_max = 180.f ;
        Lon_001:valid_min = -180.f ;
float Lon_002(along_track_002, across_track) ;
        Lon_002:long_name = "longitude for swath number 002" ;
        Lon_002:_FillValue = -999.f ;
        Lon_002:units = "degrees_east" ;
        Lon_002:valid_max = 180.f ;
        Lon_002:valid_min = -180.f ;
float Lon_003(along_track_003, across_track) ;
        Lon_003:long_name = "longitude for swath number 003" ;
        Lon_003:_FillValue = -999.f ;
        Lon_003:units = "degrees_east" ;
        Lon_003:valid_max = 180.f ;
        Lon_003:valid_min = -180.f ;
float Longitude(west_east) ;
        Longitude:long_name = "longitude" ;
        Longitude:_FillValue = -999.f ;
        Longitude:units = "degrees_east" ;
        Longitude:valid_max = 180.f ;
        Longitude:valid_min = -180.f ;
float Optical_Depth_Land_And_Ocean_001(along_track_001, across_track) ;
        Optical_Depth_Land_And_Ocean_001:long_name = "aerosol optical depth for swath number 001" ;
        Optical_Depth_Land_And_Ocean_001:_FillValue = -999.f ;
        Optical_Depth_Land_And_Ocean_001:units = "dimensionless" ;
        Optical_Depth_Land_And_Ocean_001:coordinates = "Lon_001 Lat_001" ;
float Optical_Depth_Land_And_Ocean_002(along_track_002, across_track) ;
        Optical_Depth_Land_And_Ocean_002:long_name = "aerosol optical depth for swath number 002" ;
        Optical_Depth_Land_And_Ocean_002:_FillValue = -999.f ;
        Optical_Depth_Land_And_Ocean_002:units = "dimensionless" ;
        Optical_Depth_Land_And_Ocean_002:coordinates = "Lon_002 Lat_002" ;
float Optical_Depth_Land_And_Ocean_003(along_track_003, across_track) ;
        Optical_Depth_Land_And_Ocean_003:long_name = "aerosol optical depth for swath number 003" ;
        Optical_Depth_Land_And_Ocean_003:_FillValue = -999.f ;
        Optical_Depth_Land_And_Ocean_003:units = "dimensionless" ;
        Optical_Depth_Land_And_Ocean_003:coordinates = "Lon_003 Lat_003" ;
float apcp(time, north_south, west_east) ;
        apcp:long_name = "accumulated precipitation" ;
        apcp:_FillValue = -999.f ;
```

```
                apcp:units = "kg per m^2" ;
                apcp:coordinates = "Longitude Latitude" ;
        float hpbl(time, north_south, west_east) ;
                hpbl:long_name = "planetary boundary layer height" ;
                hpbl:_FillValue = -999.f ;
                hpbl:units = "m" ;
                hpbl:coordinates = "Longitude Latitude" ;
        float psfc(time, north_south, west_east) ;
                psfc:long_name = "surface pressure" ;
                psfc:_FillValue = -999.f ;
                psfc:units = "hPa" ;
                psfc:coordinates = "Longitude Latitude" ;
        float ptrp(time, north_south, west_east) ;
                ptrp:long_name = "tropopause pressure" ;
                ptrp:_FillValue = -999.f ;
                ptrp:units = "hPa" ;
                ptrp:coordinates = "Longitude Latitude" ;
        double time(time) ;
                time:long_name = "Model Time" ;
                time:units = "hours since 2013-11-09 12:00:00 GMT" ;
        float uwind500(time, north_south, west_east) ;
                uwind500:long_name = "zonal wind at 500 mb" ;
                uwind500:_FillValue = -999.f ;
                uwind500:units = "m/s" ;
                uwind500:coordinates = "Longitude Latitude" ;
        float uwind700(time, north_south, west_east) ;
                uwind700:long_name = "zonal wind at 700 mb" ;
                uwind700:_FillValue = -999.f ;
                uwind700:units = "m/s" ;
                uwind700:coordinates = "Longitude Latitude" ;
        float uwind850(time, north_south, west_east) ;
                uwind850:long_name = "zonal wind at 850 mb" ;
                uwind850:_FillValue = -999.f ;
                uwind850:units = "m/s" ;
                uwind850:coordinates = "Longitude Latitude" ;
        float vwind500(time, north_south, west_east) ;
                vwind500:long_name = "meridional wind at 500 mb" ;
                vwind500:_FillValue = -999.f ;
                vwind500:units = "m/s" ;
                vwind500:coordinates = "Longitude Latitude" ;
        float vwind700(time, north_south, west_east) ;
                vwind700:long_name = "meridional wind at 700 mb" ;
                vwind700:_FillValue = -999.f ;
                vwind700:units = "m/s" ;
                vwind700:coordinates = "Longitude Latitude" ;
        float vwind850(time, north_south, west_east) ;
                vwind850:long_name = "meridional wind at 850 mb" ;
                vwind850:_FillValue = -999.f ;
                vwind850:units = "m/s" ;
                vwind850:coordinates = "Longitude Latitude" ;
        float zsfc(time, north_south, west_east) ;
                zsfc:long_name = "surface geopotential height" ;
                zsfc:_FillValue = -999.f ;
                zsfc:units = "gpm" ;
                zsfc:coordinates = "Longitude Latitude" ;
```

The second file contains trajectory forecast information created from running the trajectory model. It includes both the number of trajectories that were initialized as well as the location of the trajectories through the 48 hour time period. It is matched up with the gridded output NetCDF file through the xtraj and ytraj fields that contain latitude and longitude. Both NetCDF files are used to create the output file .png images. The naming convention for these files is:

MODISaerosol(X)_traj_48hr_YYYYMMDD.nc (ex: **MODISaerosolA_traj_48hr_20131110.nc**)

Where X is either A for Aqua or T for Terra. A header file dump of a typical trajectory output NetCDF file is shown below:

```
netcdf MODISaerosolT_traj_48hr_20131110 {
dimensions:
        time = UNLIMITED ; // (31 currently)
        traj = 84 ;
variables:
        float age_traj(time, traj) ;
                age_traj:long_name = "age of trajectory" ;
                age_traj:_FillValue = -999.f ;
                age_traj:units = "hours" ;
                age_traj:coordinates = "ytraj xtraj" ;
        float aod_traj(traj) ;
                aod_traj:long_name = "initial aerosol optical depth" ;
                aod_traj:_FillValue = -999.f ;
                aod_traj:units = "dimensionless" ;
                aod_traj:coordinates = "ytraj xtraj" ;
        float ptraj(time, traj) ;
                ptraj:long_name = "pressure level of trajectory" ;
                ptraj:_FillValue = -999.f ;
                ptraj:units = "hPa" ;
                ptraj:coordinates = "ytraj xtraj" ;
        float qtraj(time, traj) ;
                qtraj:long_name = "water vapor mixing ratio of trajectory" ;
                qtraj:_FillValue = -999.f ;
                qtraj:units = "kg/kg" ;
                qtraj:coordinates = "ytraj xtraj" ;
        double time(time) ;
                time:long_name = "Model Time" ;
                time:units = "hours since 2013-11-09 12:00:00 GMT" ;
        float ttraj(time, traj) ;
                ttraj:long_name = "temperature of trajectory" ;
                ttraj:_FillValue = -999.f ;
                ttraj:units = "Kelvin" ;
                ttraj:coordinates = "ytraj xtraj" ;
        float xtraj(time, traj) ;
                xtraj:long_name = "longitude" ;
                xtraj:_FillValue = -999.f ;
                xtraj:units = "degrees_east" ;
                xtraj:valid_max = 180.f ;
                xtraj:valid_min = -180.f ;
        float ytraj(time, traj) ;
                ytraj:long_name = "latitude" ;
                ytraj:_FillValue = -999.f ;
                ytraj:units = "degrees_north" ;
                ytraj:valid_max = 90.f ;
                ytraj:valid_min = -90.f ;
```

3.2.2 Output Image files

Image product files showing the hourly progression of aerosol trajectories are produced for each execution of IDEA-I MODIS aerosol. The hourly output .png format files are named xxframe.png, where xx is the hour of the forecast since 12 UTC. The last .png file will always be the 60 hour trajectory forecast, 60frame.png; the first file will depend upon the input MODIS data files and their relationship to 12 UTC of your forecast day. The size of the image files is by default 1280x960, but can be set using the IMGRES parameter in your configuration file. These files can be displayed through a web browser as an animation with controls using the PHP software that is included with this package. Please see the section on Displaying IDEA-I MODIS Aerosol Products.

If you have the ImageMajik convert command available, it will create an animated GIF file composite of all of the .png files in the directory. This file naming convention is: MODISaerosol(X)_traj_48hr_YYYYMMDD.gif (ex: MODISaerosolA_traj_48hr_20131110.gif), where X is either A for Aqua or T for Terra.

3.3 Automating Execution of IDEA-I MODIS Aerosol

Execution of the IDEA-I MODIS Aerosol Air Quality software can be automated used the unix cron scheduler. For example, to schedule a daily execution for Terra, a daily crontab would look something like this:

0 22 * * * /home/IDEA-I_aerosol/IDEA-I.bash MODIS aerosol -v -X -G -M Terra+ -d TODAY -f my_aerosol.cfg

This entry will execute at 22:00 UTC (this is dependent on your computer system time) and produce Terra trajectory products daily. If your domain is wholly covered by MODIS data from one UTC day, but the data is not all available until the next UTC day, run with –d YESTERDAY. Add the –F flag if you want to download data from the University of Wisconsin Direct Broadcast 7 day archive ftp site. To do this, you must also set your config file A04FETCH, A06FETCH, T04FETCH and/or T06FETCH parameters to YES.

30 2 * * * /home/IDEA-I_aerosol/IDEA-I.bash MODIS aerosol -v -X -G -M Aqua+ -d YESTERDAY -f my_aerosol.cfg > /home/IDEA-I_aerosol/logs/ideai_aqua_$(date +\%Y\%m\%d_\%H\%M).log 2>&1

This entry will execute the IDEA-I MODIS aerosol software for Aqua each day at 02:30 UTC and place the text output generated by standard output and standard error in a date/time stamped file in /home/IDEA-I_aerosol/logs.

Section 4: Displaying IDEA-I MODIS Aerosol Products

4.1 Displaying Aerosol Trajectories Through a Web Browser

The output image .png files can be displayed as animations in a web browser using the PHP software that is included in the IDEA-I/www directory. You must have a web server running on the computer you are serving data from.

If your web server is configured so that "http://www.your-web-site.com/" points at "/www/html/", then linking /www/html/idea-i-aerosol -> /home/IDEA-I_aerosol/www will display graphics available at http://www.your-web-site.com/idea-i-aerosol

    Example: Your IDEA-I_aerosol/www package location: ${HOME}/IDEA-I_aerosol/www

        Your web server configured: /var/www/html/

        Then link www to /var/www/html:
        cd /var/www/html
        ln -s ${HOME}/IDEA-I_aerosol/www   idea-i-aerosol

        View at: http://www.your-web-site.com/idea-i-aerosol

In the www/ directory, you can change the settings for your website IDEA-I display by editing the config.php file.

The defaults as they are set will result in any data that is located within the IDEA-I_aerosol/products/myRegion/MODISaerosol directory to be displayed with radio buttons for Terra and Aqua, and a selectable Date Calendar on the webpage. Terra is the default satellite that will appear on the website front page.

You can modify the PHP software to display different regions, satellites and products by editing the IDEA-I_aerosol/www.config.php file. Here are a few parameters that can be adjusted:

- **dataDir** - default: MODISaerosol (this directory is symbolically linked to the www directory from ../products/myRegion and contains a sample data set from Aqua and Terra over a US domain). This is the top level directory where the images being produced by your IDEA-I executions should reside. This MODISaerosol directory is currently a soft link into the IDEA-I_aerosol/www directory. The website will display a single image on the front page from the most recently processed day. By clicking on the image or on the "View Latest" button it will display the trajectory forecast imagery as an animation that you can control through a web browser. The images for an animation must be in a date sub-folder using the naming convention,

      yyyymmdd   where,

       yyyy is the year
       mm   is the month
       dd   is the day of month

      Optionally, you can use additional sub-folders to organize the date folders under the top level folder specified by "dataDir". Example:

              satellite_1
                  20120105
                      images ...
                  20120107
                      images ...
              satellite_2
                  20120105
                      images ...
                  20120107
                      images ...
    and so on ...

The date folders must be the last folders in your directory structure - they are the folders that must contain the actual images that will be animated.

- **params** - default: array('date');
      You must specify the structure of the directory tree that your images are stored in. This is done via the params array.

      Example:

      If dataDir = 'MODISaerosol';

      and the directory tree structure example shown above, then params should be

public static $params = array('satellite', 'date');

        This is specifying the structure of your directory tree where you store the images and it is also giving a parameter name to each of the directory tree levels. The last "parameter" or level in your directory tree must be your date folders (of the form yyyymmdd) and must be called date.

- **param_defaults** - default: array('latest')
        In addition to specifying the structure of the directory tree in which you store images you want to animate, you need to specify the default folder to use for each "parameter" using the param_defaults array.

        So using the example above, you could specify something like this:

        public static $param_defaults = 'satellite' => 'satellite_2',
                                        'date' => 'latest');

         This would mean that if the URL is not specifying a particular /satellite/date/ folder to animate, then the default date folder to look for images in would be

            satellite_2/ {latest}/

        where {latest} is the latest (chronologically) date folder in

            satellite_2/

- **logoFilename** - Default is my_logo.png
        You can copy your logo into the IDEA-I_aerosol/www/templates/default/images directory and it will display in the top right corner of the web page.

Other options exist that allow you to set the control buttons to be either dropdown menus or radio buttons. For a list of all the options, please see the README_IDEAI_WWW.txt file located in the IDEA-I_aerosol/www directory. The figure below is a screen capture of a website display where both MODIS Aqua and Terra are available using radio buttons for the satellite selection.
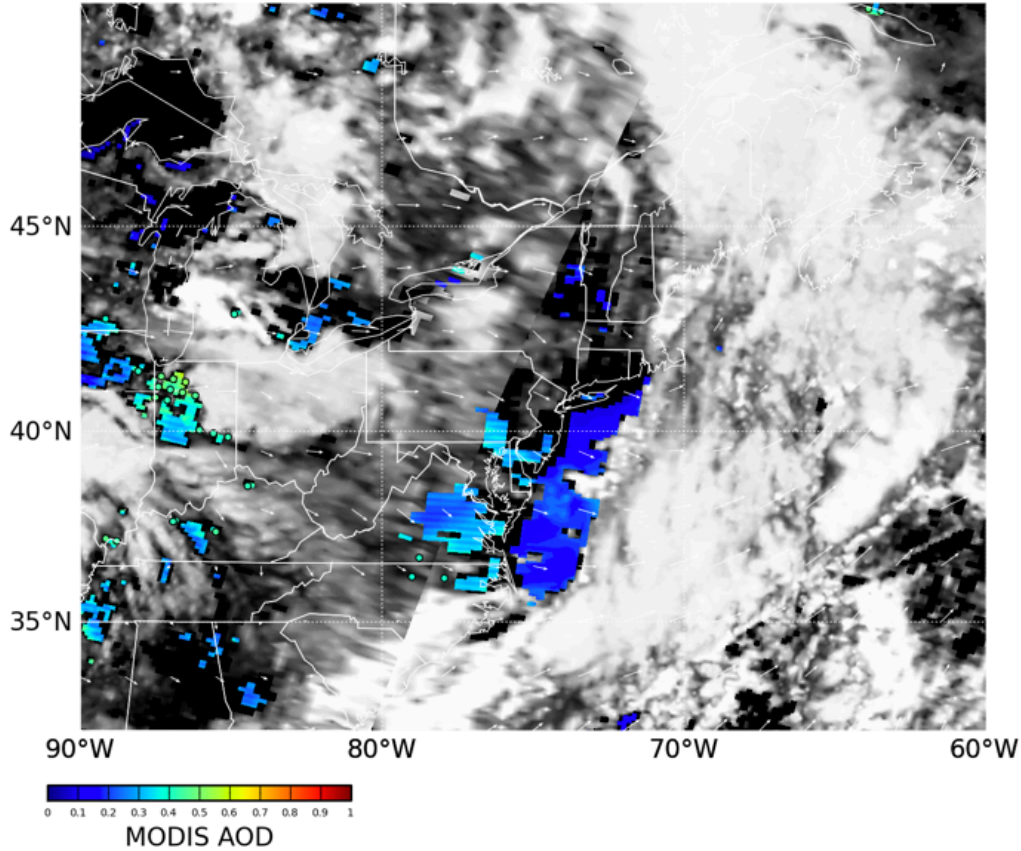
4.2 Displaying Products using McIDAS-V

A free visualization and analysis tool that can be used to display the output NetCDF gridded file is the Man computer Interactive Data Access System – V (McIDAS-V) available via download from:

http://www.ssec.wisc.edu/mcidas/software/v/

To display your IDEA-I aerosol grid files using McV, download and install the McV software onto your local machine (Linux, Mac and Windows are supported). Once the IDEA-I output gridded NetCDF file (ex: **MODISaerosolA_grid_48hr_20131110.nc**) is on your local machine, launch McV. In the Data Explorer Window, choose **Gridded Data->Local** from the left navigation options. In the **Files:** window, navigate to your NetCDF file, select the file, and click on the **Add Source** button at the bottom.   A new window will appear, with sections labeled Fields and Displays. Under **Fields**, select **2D grid**. This will expand the list of fields available for display in the file. As an example, select "aerosol optical depth for swath number 001". Once you select this parameter, the Displays section will be active. Select **Color-Shaded Plan View**, then click on the **Create Display** button at the bottom of the window. The McIDAS-V image window will display the aerosol optical depth for the first pass in the image window. If there is more than one pass stored in the file, you can display them all in the same window, by returning to the **Data Explorer-> Field Selector** window and repeating the steps taken to display the first aerosol optical depth pass.   The two following Figures show the Data Explorer->Field Selector and the McIDAS-V image window display of an IDEA-I aerosol grid file containing 3 passes captured and processed over the US from Terra MODIS on 11 November 2013. Once your data is loaded into the McV display, you can change the data range, enhancement, overlay other products, change projections, interrogate the data, etc. Please see the documentation that is available for McIDAS-V for more information:

http://www.ssec.wisc.edu/mcidas/software/v/documentation.html
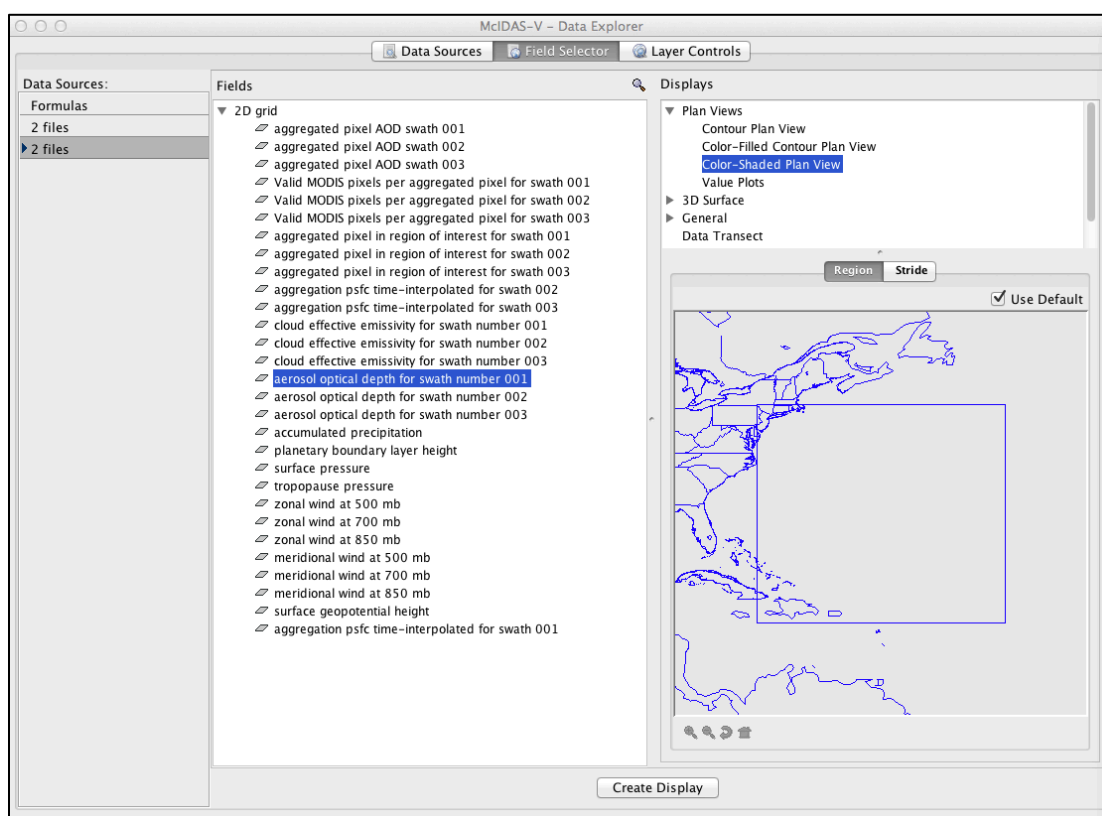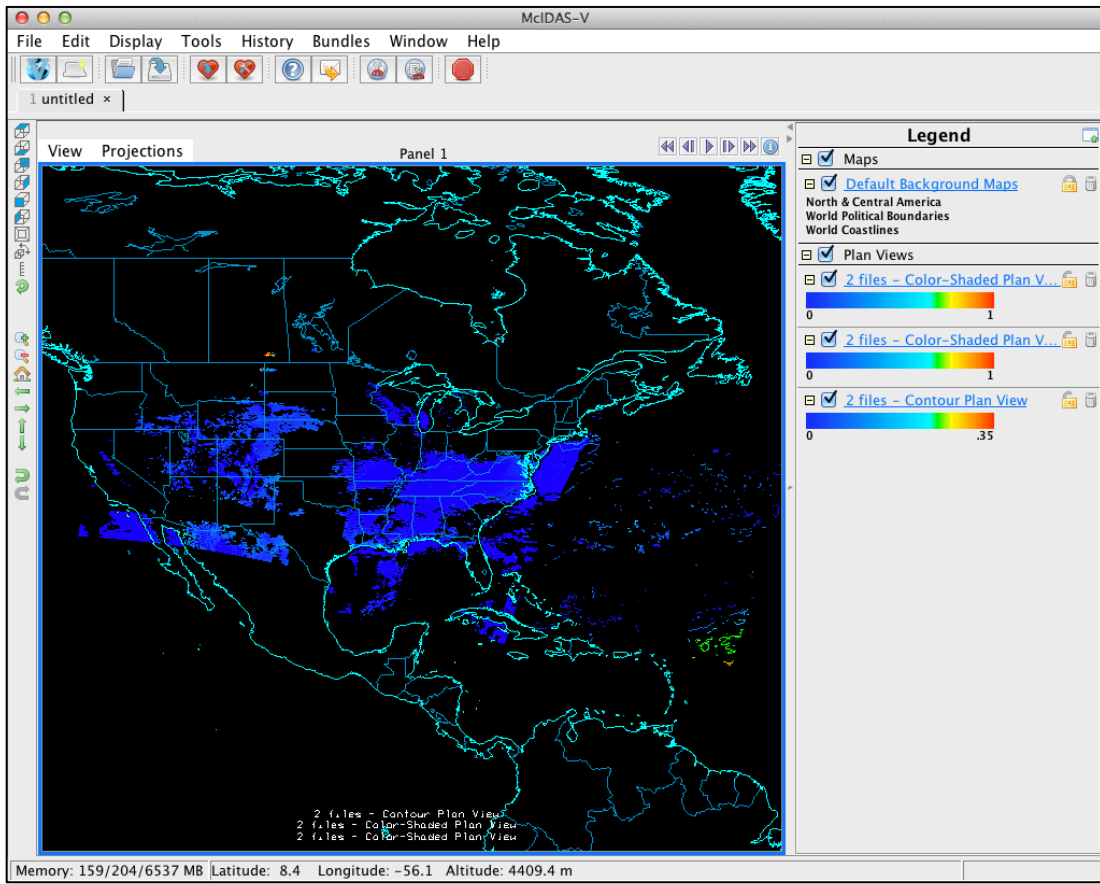
Appendix 1: IDEA-I Trajectory Model and References

A1.1 Trajectory Model

The trajectories are predicted using the NASA Langley trajectory code [Pierce and Fairlie, 1993] which solves 3 ordinary differential equations:

$$Dx/Dt = U(x,y,p,t) \qquad\qquad 1$$
$$Dy/Dt= V(x,y,p,t) \qquad\qquad 2$$
$$Dp/Dt=W(x,y,p,t) \qquad\qquad 3$$

at each time step. Here x, y, and p are the zonal, meridional and pressure of the trajectories. U and V are the zonal and meridional components of the velocity field evaluated on a pressure surface and W is the pressure vertical velocity. A 4th order Runge-Kuta implicit integration scheme [Austin and Tuck, 1985] was used to integrate equations 1-3 with three-dimensional winds which are linearly interpolated in space and time from GFS forecasts which are stored at 6 (3)-hour intervals .

A1.2 References

Al-Saadi, J. A. et al., Improving National Air Quality Forecasts with Satellite Aerosol Observations, BAMS, DOI:10.1175/BAMS-86-9-1249, 2005.

Austin, J. and A . F. Tuck, The calculation of stratospheric air parcel trajectories using satellite data., Q. J. R. Meteorol Soc., 111, 279-307, 1985.

Pierce, RB Fairlie, TDA, Chaotic Advection in the Stratosphere - Implications for the Dispersal of Chemically Perturbed air From the Polar Vortex, J. Geophys. Res., 98, 18589-18595, 1993.