DRAFT

Report on

Development of a Lagrangian Particle Dispersion Model Compatible with the Weather Research and Forecasting (WRF) Model – Phase 2

Jerome D. Fast and Richard C. Easter
Pacific Northwest National Laboratory

19 October 2006

1. Summary

During the first part of FY 2006, hereafter referred to as "phase-1", the FLEXPART Lagrangian particle dispersion model [*Stohl et al*., 2005] was adapted so that it could utilize the predicted meteorological fields produced by the Weather and Research and Forecasting (WRF) model [*Fast and Easter,* 2006a,b]. Extensive modifications of the FLEXPART code were made so that

1) the NetCDF output files produced by WRF are read directly,
2) the particle position computations correctly utilize the native WRF grid, and
3) the assumptions employed by the model are consistent with mesoscale models rather than global models.

During the second half of FY 2006, hereafter referred to as "phase-2", additional capabilities were added to the model including:

1) the ability to run the dispersion model using meteorological fields from both NCEP's Global Forecasting System (GFS) and WRF models
2) consolidated the input files into one file
3) based all documentation and coding examples in terms of latitude/longitude coordinates
4) developed IDL scripts to graphically display model output using the Interactive Data Language (IDL) software available from Research Systems Incorporated.

Because the FLEXPART code has been extensively changed, we have renamed the model the PNNL Integrated Lagrangian Transport (PILT) model. The new model is integrated in the sense that it can perform both dispersion and trajectory simulations and it can employ two-types meteorological fields.

This report summarizes the new capabilities of PILT, describes how to use the PILT for GFS and WRF applications, and discusses how the IDL scripts can be used to graphically depict predictions made by PILT. The PILT code and the associated IDL scripts should be considered a beta version. Although we have performed a number of tests, there may still be problems associated with certain configurations of the model that have not yet been checked. The IDL scripts have only been checked with the example PILT runs performed under phase-2.

2. PILT Structure

The structure of the code has changed significantly to consolidate the user input files and permit the use of both GFS and WRF meteorological fields as input. It would not have been feasible given the available resources to modify the phase-1 version of FLEXPART-WRF to accommodate the GFS model output. Fortunately a GFS-compatible version of FLEXPART by Caroline Forester recently became available on Andreas Stohl's web site, and we applied our code changes (e.g. consolidated input files, ASCII output option, etc.) to Caroline Forester's code. As a result, we have created twin versions of PILT, with the executables called *pilt_wrf* and *pilt_gfs*. The operation of these two codes is identical, with only a few minor differences as will be noted later.

All input and output are now described in terms of latitude and longitude coordinates. The WRF version of PILT still employs the native WRF coordinate system in terms of meters for computing the transport of particles, but this should be transparent to the user.

2.1 Modifications to the source code

The original top-level directory for FLEXPART-WRF (which contained source code and was also used for running the code) was divided into a "*src*" directory containing the source code and makefile, and a "*run*" directory containing input and output files as part of phase-1. As part of phase-2, the main directory now contains separate source-code sub-directories for the GFS and WRF versions of PILT, called */src_gfs* and */src_wrf*, respectively.

FLEXPART-WRF had a number of input files that have now been consolidated into one file called *pilt.input*. Information in this input file is now read by one subroutine, called *readinput.f*. This subroutine replaces *readcommand.f, readageclaseses.f, readavailable.f, readoutgrid.f readoutgrid_nest.f, readreceptors.f, readspecies.f, readreleases.f*, and *readpaths.f.* These subroutines have been removed and are no longer included in the makefile. The code in *readinput.f* assumes that *pilt.input* is in a specific format.

The consolidation of the input files has been done consistently in both the GFS and WRF versions of the code. In this way, the same *pilt.input* file can be used for both GFS and WRF applications.

2.2 Modifications to running the code

Examples of the input and output for PILT are included in the */example** sub-directories. The user needs to copy over the *pilt_gfs* or *pilt_wrf* executables into the directory in which the model is to be run.

The consolidation of the input files has also lead to a simpler structure of the output sub-directory. The */options* sub-directory is no longer needed. As a result, the 3 data input files, *landuse.asc, surfdata.t,* and *surfdepo.t* are now located at the level where PILT is run. For organizational purposes, we have retained sub-directories for the meteorological data files (from GFS or WRF) and PILT output files. The names of the meteorology data and output sub-directories are user-defined. In th examples provided, the meteorological data sub-directory is denoted by */grib* (for GFS) or */netcdf* (for WRF). Within the meteorology data sub-directory, the

user must create a file (which is named "*AVAILABLE*" in the examples provided) that lists the name and date/time for each meteorology data file to be used by PILT.

The following steps are performed to run the model:

1) Modify makefile to include the appropriate NetCDF or GRIB paths and compiler flags for your system. Compiling the code with optimization turned on and bounds checking turned off greatly increases the model speed.
2) Copy example input files to *pilt.input* and modify for specific application
3) Make sure appropriate GFS or WRF files are in either */grib* or */netcdf* and modify the *AVAILABLE* file
4) Run program by typing: *./pilt_gfs >pilt.output* for a GFS application *or ./pilt_wrf >pilt_output* for a WRF application. The standard output file, *pilt_output*, may contain useful information if the run crashes, otherwise it does not need to be saved. The most common reason the model crashes is a mismatch between the simulation times specified in the input file and the available WRF or GFS files.
5) Output will be put in a directory specified in the *pilt.input* file

2.3 The new *pilt.input* file:

The lines in the input file originate from the files *pathnames, COMMAND, AGECLASSES, OUTGRID, RECEPTORS, SPECIES,* and *RELEASES* files previously used by FLEXPART-WRF.

Examples for the pathnames section of *pilt.input* include:

GFS version:
/files1/pilt/example_gfs/output.test
/files1/pilt/example_gfs/grib
/files1/pilt/example_gfs/grib/AVAILABLE

WRF version with 1 grid:
/files1/pilt/example_wrf/output.test
/files1/pilt/example_wrf/netcdf
/files1/pilt/example_wrf/netcdf/AVAILABLE

WRF version with 3 grids:
/files1/pilt/example_wrf_nested/output.test
/files1/pilt/example_wrf_nested/netcdf
/files1/pilt/example_wrf_nested/netcdf/AVAILABLE
/files1/pilt/example_wrf_nested/netcdf
/files1/pilt/example_wrf_nested/netcdf/AVAILABLE.grid2
/files1/pilt/example_wrf_nested/netcdf
/files1/pilt/example_wrf_nested/netcdf/AVAILABLE.grid3

The first line is the path where output of PILT is written. This directory must exist prior to running the code.

The format for the *pilt.input* file is the same for both GFS and WRF applications. There are minor differences in how the file is used and they include:

1) DXOUTLON and DYOUTLAT: For GFS applications these two parameters are set to the grid distance in the x and y direction in degrees. For WRF applications, they should be set equal to the longitude and latitude of the output grid's upper-right corner, and the code will then determine the horizontal grid spacing (in internal grid-meter units) based on how many points there are as defined by NUMXGRID and NUMYGRID. The reason for this are the different native coordinate systems for GFS (lat/lon) and WRF (several types of projections). For the GFS run, the output grid is a regular grid in lat/lon space. For the WRF run, the output grid is a regular grid in the map projection space, and thus is not a regular grid in lat/lon space.

2) SFC_OPTION: For GFS applications this parameter should be set to 0. By default, PILT-GFS computes u_*, surface heat flux, and PBL height based on the mean meteorological input files. Mesoscale models, such as WRF, usually compute these variables and it would be more consistent to use the values predicted by WRF. Setting this parameter to 1 makes the code read the variables directly from the WRF output files. For this case, the user must check the WRF registry so that u_*, surface heat flux, and PBL height are written to the NetCDF output files. If PILT-WRF cannot find these variables, the code will then assume they are missing and compute u_*, surface heat flux, and PBL height by the default methodology. NOTE: if using SFC_OPTION = 1 make sure that the first WRF output file is not used since u_*, surface heat flux, and PBL height are not defined in the wrfout_* files. NOTE: for the example runs provided, WRF was run without writing u_* to the output files so SFC_OPTION was set to 0.

3) ADD_SFC_LEVEL: This option is still not fully functional yet and it should be set to 1. The purpose of this option is described in the future developments section.

4) Other internal differences in code: In PILT-WRF the parameterization of mesoscale wind fluctuations have been turned off since WRF usually explicitly predicts these quantities. For large-scale WRF simulations (Δx = ~50 km or greater) it may be appropriate to parameterize the mesoscale wind fluctuations so the lines containing *usigold*=0, *vsigold*=0, and *wsigold*=0 should then be commented out. For PILT-GFS, the code parameterizes the mesoscale wind fluctuations by default.

3. PILT Tests

We have used both GFS and WRF simulations of circulations over central Mexico for July 10 2006 to perform several tests the trajectory, particle dispersion, and concentration field computations produced by PILT. This region was chosen because of the complex terrain. While the GFS simulation encompasses the entire globe, the domain of the WRF simulation has three regional grids. The outer grid encompassed Mexico with a grid spacing of 22.5 km while inner nested grids were centered over Mexico City with grid spacings of 7.5 and 2.5 km. The boundary conditions for WRF were based on the GFS analyses for the same time period.

Input and output files are included in the */example_gfs*, */example_wrf,* and */example_wrf_nested*. Two WRF tests were performed to illustrate the differences of including higher resolution winds on transport and diffusion. One can also compare how the different wind fields from the GFS and WRF simulations affect dispersion, assuming the same release characteristics.

In each sub-directory are example input files for forward/backward dispersion for continuous (18-h) and "puff" (1-h) releases as well as input files for forward/backward trajectories. The files are named:

pilt.input.fwd:	forward dispersion
pilt.input,bwd:	backward dispersion
pilt.input.trj.fwd:	forward trajectory
pilt.input.trj.bwd:	backward trajectory

The dispersion runs are set up for an 18-h continuous release. To produce the "puff" examples, the release time was shorted to 1-h duration.

Plots of the predicted particles and concentrations from the example tests are shown in Figures 1-3. The GFS wind fields transport particles emitted from Mexico City towards the west over a 15-h period where the continuous release started by 00 UTC on July 10, 2006. The WRF winds are more southeasterly so that the particle plume is transported towards the northwest. There are also differences in the dispersion between the 1-grid and 3-grid WRF simulations. The winds from the 3-grid WRF simulation contain local variations influenced by the topography and create a more complex dispersion pattern.

4. Graphics using IDL scripts

An IDL script, called *pdis.pro* (i.e. "plot dispersion"), was written to plot particle locations, concentration fields, and individual trajectories. Another script, called *prec.pro* (i.e. "plot receptor") was written to plot a time series of concentrations at receptor locations.

pdis.pro reads and plots output (e.g. partposit*, grid_conc*, grid_time*, trajectories.txt) from both the GFS and WRF versions of PILT. The script has three panels: a horizontal cross section and two vertical cross sections that are defined by the user. While the script is long (~1500 lines) it is organized so that the three cross-section plots are divided into separate sections of code. Within each section are conditional statements that control what is being plotted. The script length results from the number of user-parameters and conditionals that make the script usable for a wide range of plot types. We cannot anticipate all of the user needs, but the organization of the script should make it relatively easy for the user to modify.

The user controls the type of plot and characteristics of the plot using a series of parameters defined at the top of the script:

```
colortype=0      ; 0 = default color table, > 0 user specified
outtype=0        ; 0 = plot to screen, 1 = plot to postscript file
```

```
saveframe=0       ; 0 = do not save *gif image, 1 = save *gif image for each frame
animate=0         ; 0 = no animation, 1 = animate
plottype=1        ; 1 = particles, 2 = concentrations, 3 = both, 4 = trajectories
model=1           ; 1 = GFS, 2 = WRF
ldirect= 1        ; -1 = backward dispersion, 1 = forward dispersion
;
lats=17.0         ; sw corner latitude of plot
latn=27.0         ; ne corner latitude of plot
lonw=-105         ; sw corner longitude of plot
lone=-95          ; ne corner longitude of plot
klevel=1          ; vertical level for concentration plots
xs_lat=20.0       ; e/w cross section latitude
xs_lon=-99.5      ; n/s cross section longitude
;
ntime=1           ; number of frames to plot – for particle and concentration plots only
ntime_traj=18     ; number of trajectory times – for trajectory plots only
dtmin=60          ; increment of particle/concentration output files in minutes
dttraj=180        ; increment of trajectory positions to plot
iyear=2006        ; beginning year
imon=7            ; beginning month
iday=10           ; beginning day
ihr=15            ; beginning hour
imin=00           ; beginning minute
isec=00           ; beginning second
;
iplot_rec=1       ; = 0 to not plot receptor sites, = 1 to plot receptor sites
numreceptors=7    ; number of receptor sites to plot
numspecies=1      ; number of species
;
fsize_tit=1.4     ; fontsize for labels
fsize_lab=1.0     ; fontsize for labels
lthick=1.0        ; line thickness
pthick=2.0        ; plot thickness
mthick=3.0        ; map thickness
ssize_part=0.3    ; symbol size for particles
ssize_traj=1.0    ; symbol size for trajectory positions
ssize_rec=1.0     ; symbol size for receptor sites
;
zplev1=0.05       ; vertical levels AGL for color-coding particles
zplev2=0.50
zplev3=1.00
zplev4=1.50
zplev5=2.00
zplev6=2.50
zplev7=3.00
```

```
zclev1=0.00          ; concentration contours
zclev2=0.05
zclev3=1.00
zclev4=2.00
zclev5=5.00
zclev6=10.0
zclev7=20.0
zclev8=50.0
zclev9=100.0
;
; location of particle/concentration/receptor files
IF(model EQ 1) THEN path='output.gfs.fwd.cont/'
IF(model EQ 2) THEN path='output.wrf.fwd.cont/'
IF(model EQ 2) THEN BEGIN
  nd_wrf=1           ; ;which WRF domain to use for topography
  IF(nd_wrf EQ 1) THEN nx_wrf=69
  IF(nd_wrf EQ 1) THEN ny_wrf=69
  IF(nd_wrf EQ 2) THEN nx_wrf=63
  IF(nd_wrf EQ 2) THEN ny_wrf=63
  IF(nd_wrf EQ 3) THEN nx_wrf=87
  IF(nd_wrf EQ 3) THEN ny_wrf=87
ENDIF
IF(xs_lon LT 0.0) THEN xs_lon=xs_lon+360.0
```

The user must select either GFS or WRF output using the parameter *model*. While the format of the particle and concentration files for the GFS and WRF versions of the code are the same, how the results are plotted needs to be treated differently. First, the script is writing to plot topography used by the meteorology input (either GFS or WRF). Second, the concentration grids are usually not the same as the native GFS or WRF grids so interpolation of the 3-D concentrations is necessary for the vertical cross section plots. The interpolation is different for the GFS and WRF applications since the GFS version of PILT has constant lat/lon horizontal grid spacing while the WRF version has constant spacing in map projection space and thus non-constant lat/lon spacing.

The topography for the GFS is read from a file called "*dump.gfs.hgt.sfc*" which was extracted from the Grib output. The topography for WRF is read directly from *wrfout** files, so the directory where the plotting script is run must contain links to the *wrfout** files. The dimensions of the 2-D topography arrays are specified in the script file using *nd_wrf, nx_wrf*, and *ny_wrf*. We were not able to figure out how to extract *nx_wrf* and *ny_wrf* directly from the wrfout* files using the NetCDF commands in IDL (this could be fixed in the future).

The plot area of the horizontal cross-section panel is defined bounded by north and south latitudes and east and west longitudes using the parameters, *lats, latn, lonw,* and *lone*. For the horizontal plot, user can define the vertical level of the concentration with the parameter *klevel*. *klevel*=1 to plot surface concentrations. The concentration grid is a terrain-following coordinate so that the horizontal plot depicts concentrations above ground level rather than at a constant

altitude. For particle plots, the entire particle field is displayed in the horizontal cross section panel.

The N/S and E/W vertical cross-section are chosen by longitude and latitude with the parameters *xs_lon* and *xs_lat*. The N/S and E/W cross section needs to pass through the GFS or WRF domain for the particle plots to work and needs to pass through the 3-D concentration grid for the concentration plots to work. The location of the N/S and E/W cross-sections are also denoted on the horizontal cross-section panel.

It is important to note that the scripts use positive longitude (0 – 360 degrees) to make interpolating concentrations easier and have some consistency between the global GFS model and regional WRF model. For the western hemisphere, the user can enter either the negative west longitude values or positive values greater than 180 for parameters such as *xs_lon, lonw*, and *lone*. The script will convert the negative longitude to positive values.

The user can plot one frame or multiple frames for animation. It is up to the user to make sure the appropriate output files are available and are consistent with the use of the parameters: *path, ntime, dtmin, iyear, imon, iday, iyr, imin,* and *isec.* IDL can be used to animate particle or concentration fields. Another useful animation option is to save each frame as a GIF file and then animate them using another software package, such as Window Media Player or Quicktime. Creating a movie using either software is quicker when viewing the results multiple times since it bypasses the need to regenerate the plot via IDL.

There are several parameters that adjust the appearance of the plot including those for line and map thickness, size of labels and symbols, and size of particles. If concentrations were computed at receptors sites, the locations can also be overlaid on the horizontal cross-section panel plot. The contour levels are also a user-defined parameter that probably needs to be modified for each application since the concentrations depend on the release strength and duration.

For trajectory plots, the length of the trajectory is determined by the parameters *ntime_traj* and *dtmin. ntime* should be set to 1, since a trajectory plot will plot all the particle positions at one time. Large dots denoting particle positions at interval *dttraj* will be plotted if *dttraj* greater than 0. Otherwise just a line will be drawn. Positions for the trajectory plots are obtained from the file "trajectories.txt". For a particle dispersion run (TURB_OPTION = 1 in *pilt.input*), the trajectory positions in trajectories.txt are the center of mass of a large number of particles. For trajectory runs of PILT, trajectories.txt will just contain the trajectories of the number specified by NUMPOINT in *pilt.input* (with TURB_OPTION = 0 and IOUT = 4). It would also be possible to read the trajectory positions from the partout* files, but it was easier just to obtain this information from one file.

Since topography is already read from the wrfout* files using NetCDF command, it is possible to also read in the wind components and other meteorological fields from WRF that could be overlaid on the dispersion fields. For example, overlaying the horizontal winds as arrows or wind barbs would help interpret the dispersion patterns. This feature has not been incorporated yet.

The script file to plot the concentrations at receptor locations, *prec.pro*, is much shorter since it designed to only produce a time-series line plot. Similar to *pdis.pro*, the user specifies information on the start time, time increment, and path where the receptor files are located. The parameter, *numreceptors* must be equal to the number of receptors in the simulation. The user can plot all of the receptor concentrations by setting *preceptor* = 0 or plot one receptor concentration when preceptor is non-zero. Currently, the plot uses a log-scale to define receptor concentrations since a wide range of concentrations can be produced. The user can change the log-scale to a linear one and the range of the y-axis.

5. Future Developments

PILT is now significantly different in many respects to the FLEXPART model it is based on. If there are updates to FLEXPART, it may be problematic to incorporate them into PILT. We would first have to determine whether any advances in FLEXPART would be of benefit to PILT. Conversations with Andreas Stohl, however, indicate that new significant developments are not expected in the near future.

There remain several issues that should be addressed for the long-term use of PILT including:

Client-specific needs: While it is relatively straightforward to set up and run a single PILT simulation, editing and running the code may become tedious for multiple applications. The consolidation of the input files now makes it easier to execute multiple PILT runs. A script program file could be created to generate a number of *pilt.input* files. For example, the user may want to perform a series of runs using the same release site with different release durations and times. A rough outline of such a script program file is as follows:

1) User modifies script program file for range of parameters to be simulated
2) Execute script program
2) Script program reads the default *pilt.input* file (or parameters defined internally)
3) Script program creates multiple input files (e.g. *pilt.input.test1*, *pilt.input.test2*, etc.) based on user choices
4) Script program performs a series of simulations sequentially by each *pilt.input.test** file to pilt.input and then running PILT.

There are likely to be other options of interest to the client that would involve further modification of the code.

Turbulence: It would be useful to have the turbulent components consistent with WRF turbulence kinetic energy or eddy exchange coefficients. A placeholder has been set in the COMMAND file using the TURB_OPTION=2 option, but this option should not be used.

Other Graphical Display Options: We have provided on IDL script that can read the ASCII output of PILT and display the results. Another way of graphically depicting model output is through the use Google Earth. The HYSPLIT web site now provides Google Earth as one option of viewing the output. In addition to providing satellite images as the background, Google Earth

also permits a more flexible graphical user interface to view particle trajectories and concentrations at any angles that include 3-D perspectives of topography and building.

Testing of Input Options: We have not had tested all possible combinations of input options. Some of our tests have shown that some combination of options do not work as advertised. For example, we had problems using a time averaging period of less than one hour. Additional tests are needed to identify combinations of input options that do not work and possibly providing fixes for them.

Near-surface meteorological quantities: FLEXPART uses 10-m wind and 2-m temperature and humidity information to define an additional shallow vertical layer (adjacent to the ground) that is used in dispersion calculations. The default methodology is useful for global model output that has coarse vertical grid spacing. For a mesoscale model such as WRF, the vertical grid spacing may be small so that meteorological quantities are already computed near the ground. We have set a placeholder in the *pilt.input* file by setting ADD_SFC_LEVEL = 0 to deactivate the interpolation of surface quantities and rely on WRF predicted variables, but this option is not function and it should not be used.

Map factors: Particle positions and motions are calculated using the WRF map projection grid. Map factors should be (but are not yet) included in these calculations. For simulation domains covering a few thousand km, the map factors are close to unity (especially for the Lambert conformal projection), and neglecting the map factors should have little effect. Bigger errors could occur with larger simulation domains.

References

Fast, J.D., and R.C. Easter, 2006: Report on Development of a Lagrangian Particle Dispersion Model Compatible with the Weather Research and Forecasting (WRF) Model. February 16, 2006.

Fast, J.D., and R.C. Easter, 2006: A Lagrangian particle dispersion model compatible with WRF. *7th WRF Users' Workshop,* NCAR, June 19-22, Boulder, CO, p6.2.

Stohl, A., C. Forster, A. Frank, P. Seibert, and G. Wotawa, 2005: Technical Note: The Lagrangian particle dispersion model FLEXPART version 6.2. *Atmos. Chem. Phys. Disc*., 5. 4739-4799.

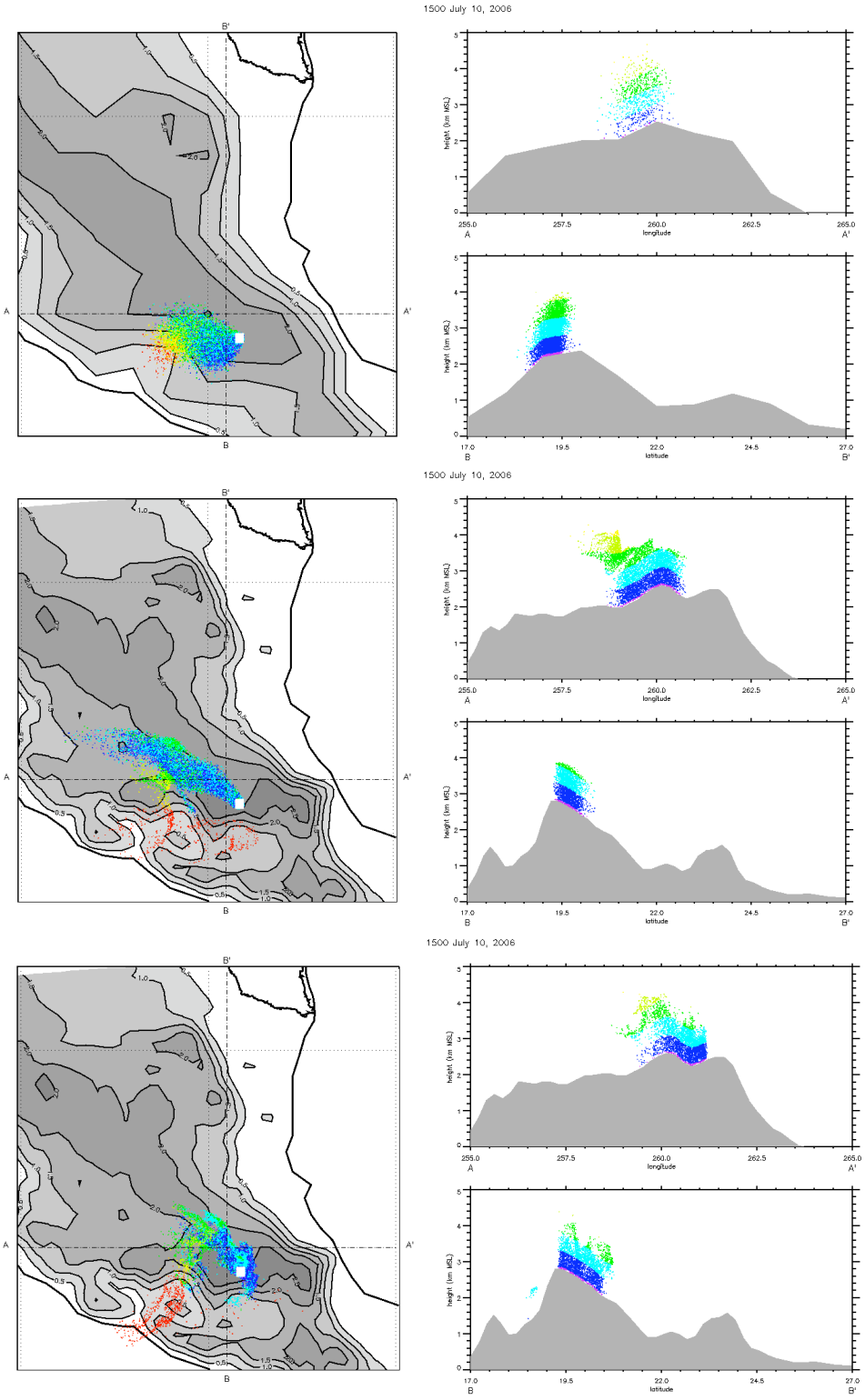

Figure 1. Predicted forward particle dispersion from GFS (top), 1-grid WRF (middle), and 3-grid WRF (bottom) simulations. Particles continuously released from Mexico City (denoted by white box) and results shown 15-h after the initial release time at 15 UTC July 10, 2006. Vertical cross-sections depict particles within 0.25 degree latitude or longitude of the vertical plane. Colors denote height above ground level.

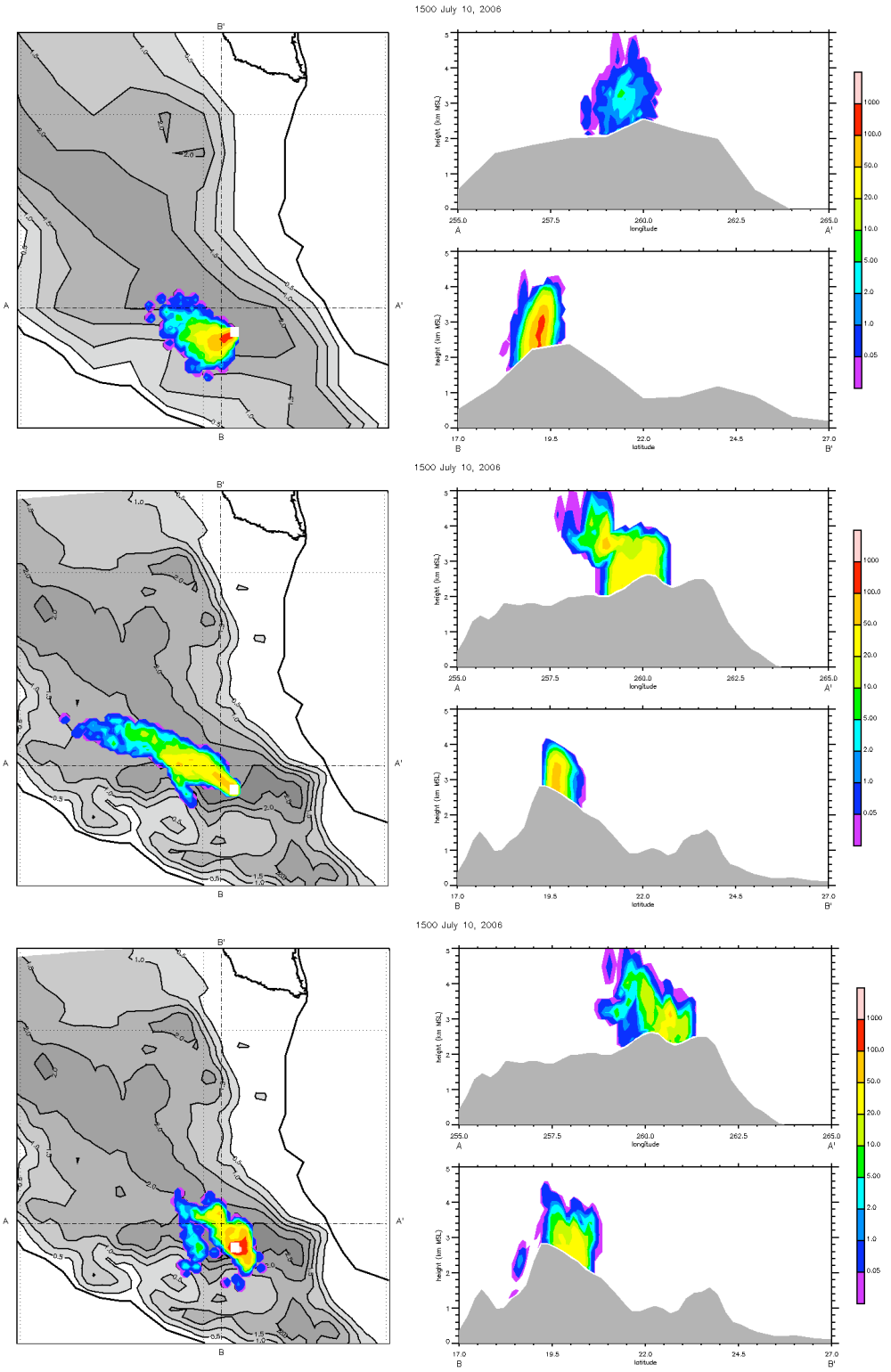

Figure 2. Same as Figure 1, except that output concentrations are plotted rather than particles.

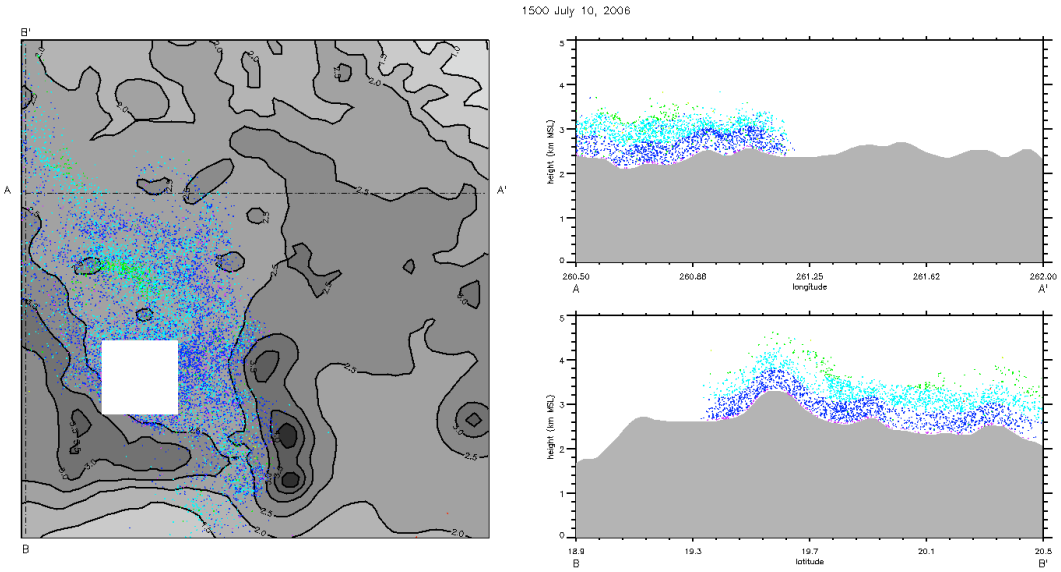

Figure 3. Same as bottom panel of Figure 1, except plotting particles close to the source and using the higher-resolution topography used by the grid 3 in WRF. White box denotes release area.