CHARACTERIZING THE AIR QUALITY AND DEMOGRAPHIC IMPACTS OF AIRCRAFT EMISSIONS AT THE HARTSFIELD-JACKSON ATLANTA INTERNATIONAL AIRPORT

Jeffrey Rissman

A thesis submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Master of Science in the Department of Environmental Sciences and Engineering and the degree of Master of City and Regional Planning in the Department of City and Regional Planning.

Chapel Hill

2011

Approved by:

Saravanan Arunachalam

J. Jason West

Todd BenDor

ABSTRACT

JEFFREY RISSMAN: Characterizing the Air Quality and Demographic Impacts of Aircraft Emissions at the Hartsfield-Jackson Atlanta International Airport
(Under the direction of Saravanan Arunachalam, J. Jason West, and Todd BenDor)

This study examined the impacts of aircraft emissions on fine particulate matter ($PM_{2.5}$) concentrations during the months of June and July 2002 at the Hartsfield-Jackson Atlanta International Airport. Pollutants were modeled using the Advanced Modeling System for Transport, Emissions, Reactions, and Deposition of Atmospheric Matter (AMSTERDAM). We also investigated the concentration changes caused by AMSTERDAM's plume-in-grid process. A geographic information system was used to apportion pollutant concentrations to census tracts in the Atlanta area. Aircraft impact on $PM_{2.5}$ was compared with demographic variables to evaluate whether minority or low-income residents are disproportionately exposed to aircraft emissions. Aircraft increase average $PM_{2.5}$ concentrations by up to 235 ng/m^3 near the airport and by 1-7 ng/m^3 throughout the Atlanta metro area. Census tracts with high aircraft $PM_{2.5}$ contribution disproportionately have low-income and minority residents. Aircraft contribution to $PM_{2.5}$ may have caused approximately 1.4 premature adult deaths in 2002.

ACKNOWLEDGEMENTS

This work was funded by PARTNER, a joint project of the FAA, NASA, EPA, DoD, and Transport Canada. The Investigation of Aviation Emissions Air Quality Impacts project is managed by Christopher Sequeira. Thank you for supporting this study.

We wish to acknowledge Hsi-Wu Wong of Aerodyne, Inc. and Francis Binkowski of the UNC Institute for the Environment for providing alternative measurement-based emissions data for elemental carbon and sulfate. We acknowledge the Georgia Department of Natural Resources for providing meteorology data, and we acknowledge Prakash Karamchandani and Krish Vijayaraghavan for providing the AMSTERDAM model code. We would like to thank Matt Woody, Lawrence Rissman, Neil Davis, and Mark Reed for their support and advice regarding code-related issues and tools.

Any opinions, findings, conclusions, or recommendations are those of the authors and do not necessarily reflect the opinions of PARTNER or its sponsors.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADSC	Aerosol Dynamics Simulation Code
AEDT	Aviation Environmental Design Tool
AERMOD	American Meteorological Society / U.S. EPA Regulatory Model
AMSTERDAM	Advanced Modeling System for Transport, Emissions, Reactions, and Deposition of Atmospheric Matter
APT	Advanced Plume Treatment
ATL	Hartsfield-Jackson Atlanta International Airport
CB-IV	Carbon Bond 04 chemical mechanism
CMAQ	Community Multiscale Air Quality modeling system
CSN	Chemical Speciation Network
EDMS	Emissions and Dispersion Modeling System
ICAO	International Civil Aviation Organization
LTO	Landing and Take Off
MADRID	Model of Aerosol Dynamics, Reaction, Ionization, and Dissolution
PARTNER	Partnership for AiR Transportation Noise and Emissions Reduction
PinG	Plume-in-Grid

$PM_{2.5}$	Fine Particulate Matter (aerodynamic diameter 2.5 microns or less)
POA	Primary Organic Aerosol
RR	Relative Risk
SCICHEM	Second-order Closure Integrated Puff Model with Chemistry
SCIPUFF	Second-order Closure Integrated Puff Model
SOA	Secondary Organic Aerosol
TC	Test Case
TOG	Total Organic Gas

I. A Plume-in-Grid Approach to Characterize Air Quality Impacts of Aircraft Emissions at the Hartsfield-Jackson Atlanta International Airport

Chapter Abstract

This study examined the impacts of aircraft emissions on $PM_{2.5}$ concentrations during the months of June and July, 2002 at the Hartsfield-Jackson Atlanta International Airport. Primary and secondary pollutants were modeled using the Advanced Modeling System for Transport, Emissions, Reactions, and Deposition of Atmospheric Matter (AMSTERDAM). AMSTERDAM is a modified version of the EPA's Community Multiscale Air Quality (CMAQ) model that incorporates a plume-in-grid process to simulate emissions sources of interest. Three fundamental issues were investigated: the effects of aircraft on $PM_{2.5}$ concentrations throughout northern Georgia, the differences resulting from use of AMSTERDAM's plume-in-grid process rather than a traditional CMAQ simulation, and the differences resulting from the use of alternative emissions inputs in which black carbon emissions are based on measurements from three measurement campaigns. Aircraft increase average $PM_{2.5}$ concentrations by up to 235 ng/m^3 near the airport and by 1-7 ng/m^3 throughout the Atlanta metro area. Aircraft decrease sulfate aerosol concentrations in rural areas by up to 1 ng/m^3. The plume-in-grid process increases concentrations of secondary PM pollutants by 0.5-20 ng/m^3 but tends to reduce the concentration of non-reactive primary PM pollutants by up to 13 ng/m^3, with changes concentrated near the airport. Use of a measurement-based

emissions index for elemental carbon increases modeled concentrations by up to 212 ng/m^3 but also increases model error and bias at an air quality monitor location in the vicinity of the airport.

1. Introduction

The Hartsfield-Jackson Atlanta International Airport is the busiest airport in the world in terms of passenger traffic (Airports Council International, 2010). Major airports such as Atlanta Hartsfield can be important economic drivers in their regions and key transit hubs for people worldwide.

However, a large airport produces significant emissions that have the potential to adversely affect air quality in communities near the airport and throughout the wider region. Aircraft engines emit nitrogen oxides (NO_x), sulfur dioxide, black carbon (soot), and other compounds that are primary air pollutants or contribute to formation of secondary pollutants through chemical reactions in the atmosphere.

In this work, the Advanced Modeling System for Transport, Emissions, Reactions, and Deposition of Atmospheric Matter (AMSTERDAM; Karamchandani et al., 2010) was used to evaluate aircraft impacts on ground-level pollutant concentrations during June and July 2002 at the Atlanta Hartsfield airport. Our study had three primary objectives. First, we wished to characterize the effect of aircraft emissions at Hartsfield-Jackson airport on ground-level fine particulate matter ($PM_{2.5}$) concentrations in Atlanta and surrounding areas. Second, we hoped to understand the differences in modeled $PM_{2.5}$ concentrations resulting from the use of the AMSTERDAM model, which incorporates a plume-in-grid process for aviation emissions, relative to a the use of a traditional gridded

air quality model. Finally, we sought to compare $PM_{2.5}$ results between model runs in which black carbon emissions were calculated using a first-order approximation and results in which these emissions were primarily based on measurement values.

For purposes of this paper, we examine only total $PM_{2.5}$ and its components (black carbon, sulfate, nitrate, ammonia, organic, and uncategorized $PM_{2.5}$). We limit our present discussion to $PM_{2.5}$ because of the significant potential health impacts of fine particulate matter (Levy et al., 2008). Exposure to $PM_{2.5}$ has been linked to increased rates of lung cancer and cardiopulmonary mortality (Pope et al., 2002). An analysis consolidating the views of 12 health experts who have conducted research on $PM_{2.5}$ exposure and mortality estimated a reduction of 1 $\mu g/m^3$ in annual average $PM_{2.5}$ concentration reduces the annual adult mortality rate by 0.7% – 1.6% (Industrial Economics, 2006).

Knowledge of aircraft contribution to pollutant concentrations is an important first step in understanding the human and economic costs associated with aircraft-related pollution. Such data may also provide a useful baseline for comparison with scenarios which consider the efficacy of mitigation measures, such as use of biofuels, altering airline flight schedules, re-balancing load between different regional airports, and other techniques. Results may also be of interest to local governments siting new airports and to regulators setting emissions standards for jet engines.

2. Model History and Development

Computer modeling is a technique that has been used to evaluate air quality impacts from airports for decades. Two models commonly used to quantify aircraft

emissions in the United States are the Emissions Dispersion and Modeling System (EDMS) (Federal Register, 1998) and its successor, the Aviation Environmental Design Tool (AEDT) (FAA, 2010).

EDMS is the most commonly used aircraft emissions model because in 1998 the FAA designated it as the required model for air quality analyses of aviation emissions sources. Hundreds of papers have been published using EDMS to evaluate various aspects of aviation emissions. EDMS was developed by CSSI, Inc. and calculates aviation emissions during the landing and takeoff cycle (LTO), i.e. within the lowest 3,000 feet of altitude near airports. Outputs are intended for use in the American Meteorological Society/U.S. EPA Regulatory Model (AERMOD) (Cimorelli et al., 2005). Researchers have used EDMS to address topics which relate to local air quality impacts, such as the effects of a reduction in aircraft thrust takeoff on annual NO_x emissions (Hall, 2003) and the air quality impact of new construction projects at airports (Moss, 1994). However, EDMS does not account for aviation emissions within the framework of a comprehensive, gridded, region-wide air quality model, and many studies using EDMS were conducted years ago, when scientific knowledge of atmospheric chemistry was less advanced and computer power was insufficient to model air quality with the resolution and accuracy achievable today.

The FAA is currently developing the Aviation Environmental Design Tool (AEDT), a model that estimates aircraft emissions, noise, and other impacts. EDMS is embedded within AEDT and handles emissions tasks. AEDT is not yet publicly released (FAA, 2010), but preliminary versions of this model have nevertheless been used by researchers, sometimes limited to a model assessment capacity (Noel, 2009), but

sometimes to provide emissions data to support more detailed analyses and predictions of future conditions (Wilkerson, 2010).

EDMS and AEDT are primarily tools that generate emissions inventories for use in other models (such as AERMOD). In contrast, the AMSTERDAM model used in this study takes emissions as inputs and simulates meteorology, chemistry, and physical processes to determine the resulting environmental concentrations. As AMSTERDAM was built from a combination of two earlier models, we first discuss each of those component models. The first component is the Community Multiscale Air Quality modeling system (CMAQ) (Byun and Ching, 1999; Byun and Schere 2006; CMAQ, 2011). CMAQ is an Eulerian model based on a 3-D grid. It accepts meteorology and emissions inputs, models chemical and microphysical processes, and determines the resulting chemical concentrations in each grid cell. CMAQ has been used in the past to model the impacts of aviation emissions (Unal et al., 2005; Ratliff et al., 2009; Arunachalam et al., 2008, 2011; Woody et al., 2011). However, CMAQ is limited by its reliance on grid cells that are relatively large in size and may not accurately represent the chemical processes in concentrated emissions plumes, such as those emitted by aircraft. Large grid cells may result in an artificial dilution of emissions, which can alter the chemical reactions which form pollutants and can result in the under-prediction of pollutant concentrations near emissions sources (Arunachalam et al., 2008).

The second component is the Second-Order Closure Integrated Puff Model with CHEMistry (SCICHEM), which is itself an evolution of a model called "SCIPUFF." SCIPUFF was originally developed for the military and could model the transport of hazardous materials (Sage Management, 2010). SCIPUFF is a Lagrangian dispersion

model. It tracks emissions as a series of individual "puffs" in three-dimensional space. The U.S. EPA approved a public domain version of the model, released in 2000, for use as an alternative model in regulatory applications (EPA). SCIPUFF focuses on modeling the location, size, and shape of emitted puffs. It accounts for puff movement due to winds, buoyancy, and other physical factors. It is capable of dividing puffs that become too large into smaller puffs and merging overlapping puffs together. Mass, heat, and momentum are all taken into account (Karamchandani, 2010).

SCIPUFF was later extended with the capability to utilize chemistry routines to account for reactions occurring inside the puffs. The resulting model was named "SCICHEM" (Santos et al., 2000). SCIPUFF and SCICHEM have been used in numerous research studies of plume dispersion, but they have seldom if ever been used to model aviation emissions.

Finally, we wish to review the development history behind AMSTERDAM, the model we used for the present study. In the early 2000s, Karamchandani et al. (2002) created an air quality model by combining CMAQ and SCICHEM. They called their model "CMAQ-APT" wherein "APT" stands for "Advanced Plume Treatment," referring to the capabilities added by SCICHEM.

By late 2006, Karamchandani et al. extended their model to utilize a different mechanism for simulation of particulate matter: the Model of Aerosol Dynamics, Reaction, Ionization, and Dissolution (MADRID). The new model was called "CMAQ-MADRID" (or "CMAQ-MADRID-APT" for a version also utilizing the SCICHEM

plume-in-grid process) (Karamchandani et al., 2006). The most important changes related to the model's particle size distribution and secondary organic aerosol formation.

Karamchandani et al. continued to develop the model, refining mercury- and aerosol-related processes. They produced a new version of the model called the "Advanced Modeling System for Transport Emissions, Reactions and Deposition of Atmospheric Matter" (AMSTERDAM), whose impacts on mercury results were examined in a 2008 paper (Vijayaraghavan et al., 2008). In a later paper, the same team describes AMSTERDAM as a "suite" of three models: CMAQ-AERO3-APT (which uses the original CMAQ particulate matter treatment plus SCICHEM plume-in-grid), CMAQ-MADRID (which uses the MADRID particulate matter treatment without plume-in-grid), and CMAQ-MADRID-APT (which uses both the MADRID PM treatment and SCICHEM plume-in-grid) (Karamchandani et al., 2010).

In our study using AMSTERDAM, emissions that are not of direct interest (i.e., non-aviation emissions) are termed "background emissions" and are added directly to the grid, as in a regular CMAQ model run. Emissions of interest (i.e., aviation emissions) are emitted as Gaussian puffs, which are tracked in three-dimensional space within the CMAQ grid. The chemistry routines in CMAQ are used to model reactions within the puffs as well as within each grid cell. As puffs age, they grow larger and more dilute due to infiltration of background air. When puffs are sufficiently large or dilute, it is no longer worthwhile to track them separately from the surrounding air. At that point, the puffs' contents are added to the grid cells where the puffs are located, and the puffs themselves are removed from the model. Thus, in each timestep, new puffs are being added to the simulation while old puffs are being "merged" into the grid. Using this

mechanism, AMSTERDAM allows emissions sources of interest to be modeled at a much higher spatial resolution than the surrounding grid.

3. Prior Modeling Work on Hartsfield-Jackson International Airport

This study extends prior modeling work that has been conducted to understand the impacts of aircraft emissions at the Hartsfield-Jackson Atlanta International Airport.

Unal et al. (2005) modeled aircraft emissions during a high-pollution episode from August 11-20, 2000 using CMAQ. For $PM_{2.5}$ emissions, they relied on a first-order approximation by Wayson et al. (2003), which is a predecessor to the method used in this study (Wayson et al., 2009). When using emissions differentiated by flight mode and added to the model in 3-D space based on aircraft flight paths, they found the airport had a maximum impact on $PM_{2.5}$ levels of 4.4 $\mu g/m^3$ and impacts of about 1 $\mu g/m^3$ in a radius 16 km around the airport.

In order to enable the airport to be efficiently examined with the CMAQ model, Baek et al. (2007) developed a tool, "EDMS2Inv," which is capable of converting the emissions files created by the EDMS model (intended for use in AERMOD) into a format that allows them to be used in CMAQ instead. This tool was used to conduct an analysis of the impacts of landing-and-takeoff cycle (LTO) emissions at three airports: Providence T. F. Green (PVD), Chicago O'Hare (ORD) and Atlanta Hartsfield-Jackson (ATL) (Arunachalam et al., 2008). In that study, annual and monthly average aircraft contribution to $PM_{2.5}$ and its components were modeled at CMAQ grid resolutions of 12 km and 36 km. Results were compared to measurement data from several air quality monitoring networks. Arunachalam et al. found that in the grid cell with the maximum

impact due to aircraft at the 12 km resolution, annual average $PM_{2.5}$ concentrations were increased by over 0.2 $\mu g/m^3$.

Woody and Arunachalam (2010) investigated the effects of aircraft on secondary organic aerosol (SOA) formation near Hartsfield-Jackson International Airport. That study used EDMS emissions processed through the EDMS2Inv tool as input data for CMAQ model runs at 36 km, 12 km, and 4 km grid resolutions. A 2-day period in June was modeled. Woody and Arunachalam found that in the grid cell containing the airport at the 36 km and 12 km resolutions, aircraft reduced SOA concentrations because of the reaction of aircraft NO_x emissions with NO_3 radicals, leaving fewer nitrate radicals to oxidize SOA precursors. However, at the 4 km grid resolution, aircraft increased SOA concentrations by providing a higher concentration of primary organic aerosol (POA) to serve as seed particles onto which SOA could partition. The use of larger grid cells diluted POA concentrations, lowering SOA production in those model runs.

A study estimating mortality due to $PM_{2.5}$ exposure from LTO emissions at three airports (ATL, ORD, and PVD) used emissions derived from EDMS in CMAQ simulations (Arunachalam et al., 2011). ATL was investigated at 36 km, 12 km, and 4 km grid resolutions. The run with 4 km resolution included June and July 2002 and used the same background (i.e., non-aviation) emissions as the present study. Arunachalam et al. found that aircraft contribution to $PM_{2.5}$ from the three airports was responsible for 38 premature deaths in 2002, only 64% - 72% of which occurred within 612 km x 612 km domains centered on the three airports. The study concluded that a lower-resolution model with large domain is suitable for understanding population-average exposure, but a

maximum individual risk assessment would require high-resolution modeling near the airport.

4. Modeling Approach

In this work, we completed four model runs, or "test cases," using a modified version of the CMAQ-AERO3-APT model from AMSTERDAM. All test cases used a 504 km x 408 km domain with 4 km grid cell resolution centered on the Atlanta airport. The model used 19 vertical layers whose thickness increased with altitude. Air quality was modeled for June and July 2002 (plus an 11-day spin-up period in May).

Each test case used background (non-aviation) emissions based on the National Emissions Inventory (NEI) for 2002 produced by the U.S. Environmental Protection Agency (USEPA, 2010). Meteorology data for 2002 were provided by the Georgia Department of Natural Resources. The differences between the four test cases are summarized as follows:

Test case 1 (**TC1**) included aviation emissions modeled using the plume-in-grid (PinG) process built into the AMSTERDAM model.

Test case 2 (**TC2**) omitted aviation emissions.

Test case 3 (**TC3**) included aviation emissions but avoided use of the PinG process by adding those emissions to the CMAQ grid directly. The emissions are identical in quantity and chemical composition to those in TC1.

Test case 4 (**TC4**) included an alternative set of aviation emissions, wherein the emissions indices for black carbon and sulfate from the five most commonly used

engines were obtained primarily from measurements. The emissions were modeled through the PinG process built into the model.

In TC1 and TC3, we use engine data from the International Civil Aviation Organization (ICAO) and a methodology developed by Wayson et al. (2009) to estimate sulfate, organic $PM_{2.5}$, and black carbon aviation emissions. This methodology, known as FOA3.0 (version 3.0 of a first-order approximation), is used in the EDMS and AEDT models. It is based on a statistical correlation between smoke number (a metric based on the coloration of soot from a particular engine) and available data on PM emissions by engine.

TC4 uses most of the same inputs as TC1, but with certain changes to elemental carbon and sulfate emissions. These changes relate to inputs for a different model, the Aerosol Dynamics Simulation Code, which will simulate the evolution of sulfate and black carbon particulate emissions downstream of an aircraft engine (Wong et al., 2008). Inputs used by the ADSC model include emissions indices (ratios of the mass of a pollutant emitted per kilogram of fuel burned) for particular engines. The ADSC developers provided us with the emissions indices they are using for black carbon and sulfate for the five most-commonly used engines at Hartsfield-Jackson Airport. In the case of black carbon, the ADSC emissions indices are based primarily on measurements from three field campaigns: APEX-1, APEX-2, and Atlanta (Timko et al., 2010) (Herndon et al., 2008). In TC4, we replace our black carbon and sulfate emissions indices for the top five engines with the corresponding ADSC indices when calculating speciated emissions for our model run. These five engines account for 74.3% of the total activity at Hartsfield-Jackson in 2002 (CSSI, 2007). TC4's sulfate aircraft emissions

were 99% of those in TC1, while TC4's elemental carbon aircraft emissions were 543% of those in TC1.

Our study attempts to build on prior work in a number of ways. This is the first time that a plume-in-grid model has been used to characterize aviation emissions in the context of a regional air quality model with chemistry (CMAQ). EDMS and AERMOD, the most commonly used models in the past, may not have been able to adequately represent the chemical processes occurring in the aircraft plume and may not have been as effective at predicting aircraft impacts at large distances from the airport.

In addition, this is the first use of AMSTERDAM in which multiple emitters have been used to represent emissions from a large number of individual points (aircraft engines) that vary in space and time. In the past, AMSTERDAM has primarily been used to study emissions from power plants (Karamchandani et al., 2006; Vijayaraghavan et al., 2008). These large sources were widely spread across the country, and a single Gaussian puff emitter represented each source. In this study, we used 51 distinct emitters to represent activity in the immediate geographic vicinity of the airport, and emitters were divided into four types based on the mode of flight they represented (taxi, take off, climb out, and approach). By using many emitters in close proximity to one-another with differing emissions profiles, we gain finer control over the location and characteristics of the emissions we input into the model.

Third, this work uses a new method to calculate aircraft emissions for use in the model. This method is based on a detailed understanding of the characteristics of the individual aircraft engines used at Hartsfield-Jackson Airport, as well as the number of

flights that occurred in each hour of the modeling period. This method is described in detail below, and the resulting emissions totals are compared to the EDMS-derived values used in prior work.

Fourth, the AMSTERDAM model was enhanced with improved capability to output sub-grid scale puff data, and numerous bugs were fixed. (See Appendix A for details regarding bug fixes in the AMSTERDAM model.)

5. Calculating Aviation Emissions

In order to model air quality impacts from the Atlanta International Airport, it was necessary to represent aircraft emissions as a number of point source emitters. This involves two steps: determining the types and quantities of pollutants that are emitted at each timestep and positioning the emitters in 3D space. These steps are discussed in the next two subsections.

5A) Quantifying Emissions by Timestep

In this project, we chose to include landing and take off cycle (LTO) emissions up to 3000 feet above the ground. Emissions in this altitude range can be broken down into four categories: **taxi** (or **idle**) emissions which occur on the ground, **take off** emissions which occur from zero to 1000 feet for departing planes, **climb out** emissions which occur from 1000 to 3000 feet for departing planes, and **approach** emissions from 3000 to zero feet for arriving planes (Rice, 2003).

Emissions above 3000 feet, including all **cruise** emissions, are omitted from the model runs. We chose to omit these emissions to help focus on the effect of LTO

emissions on surface air quality. (Also, cruise-phase emissions data were not available.) Omitting emissions above this altitude may lead to an under-estimation of overall aircraft impacts at large scales. In our model, this may particularly affect grid cells that are not near the airport, especially if they lie along flight corridors.

In this study, we performed our own emissions calculation based on the physical characteristics of each engine model when operating in each mode of flight (i.e. we did not use the outputs of a model). This is in contrast to prior work performed for the Atlanta airport (Unal et al., 2005; Arunachalam et al., 2008, 2011), wherein aircraft activity was based upon emissions inventories produced by EDMS. We use 2005 arrival and departure data to represent the number of flights using each engine at Hartsfield-Jackson in 2002 (the year represented by our other data files, such as meteorology and background emissions) because a detailed breakdown of how many flights used each model of engine was not available for 2002.

Aircraft engines have different emissions characteristics depending on their power setting. Key properties that change with power setting include fuel flow rate, smoke number (a metric based on the coloration of emitted soot), and emissions indices (mass ratios of pollutant emitted to fuel burned) for various pollutants. Therefore, in order to determine the total amount of each pollutant emitted by aircraft during 2005, it is necessary to total the emissions from each model of engine from each phase of the LTO cycle. This was done according to Equation 1 for each chemical species:

$$\sum_{1}^{num\_engine\_models} 2F_{engine}\left(\left(T_{takeoff} * E_{takeoff}\right) + \left(T_{climbout} * E_{climbout}\right) + \left(T_{approach} * E_{approach}\right) + \left(T_{taxi} * E_{taxi}\right)\right) \quad [1]$$

In Equation 1, F_{engine} refers to the number of flights using that engine model in 2005. T_{mode} is time-in-mode, the average number of seconds an aircraft spent in that particular mode of flight. E_{mode} refers to the emissions rate of a pollutant (in g or mol / s) for that particular engine model in that flight mode. The coefficient 2 is present because we assume that there are approximately two engines per aircraft. Thus, we sum the total emissions from every engine in each mode to achieve a final, pollutant-specific total.

The values of F_{engine} for each engine model were obtained from CSSI (2007). For time-in-mode values for the three flight modes ($T_{takeoff}$, $T_{climbout}$, and $T_{approach}$), we chose to use the reference values recommended for gaseous emissions calculations by the International Civil Aviation Organization (ICAO, 1993). These are: 0.7 minutes for take off, 2.2 minutes for climb out, and 4.0 minutes for the approach. Although the accuracy of these values for modern jet aircraft has been questioned (Rice, 2003), we felt there were no more accurate, authoritative values available. In contrast, CSSI calculated an average value for taxi mode specific to the Atlanta International Airport in 2005 (T_{taxi} = 27.22 min). This value was used instead of ICAO's reference value of 26.0 minutes.

The emissions rate values (E_{mode}) were calculated according to the following methodology. Key engine characteristics necessary to calculate emissions rates were obtained from ICAO's engine databank (2010). The engine databank provided fuel flow rates (kg/s), engine type (single or multiple turbofan), bypass ratio (ratio of air drawn through the periphery of the engine to air drawn through the core), smoke numbers, and emissions indices for hydrocarbons, carbon monoxide, and NO_x. When an engine did not have mode-specific smoke number data available in the ICAO databank, the Calvert method was used to calculate mode-specific smoke numbers (Eyers, 2007).

Emission indices for SO_2, H_2SO_4, organic PM, and elemental carbon (also called black carbon or soot) were obtained via the methodologies described by Wayson et al. (2009). However, for some values related to sulfur, we used numbers suggested by the FAA for use in the AEDT model. In particular, we assume 2% of emitted sulfur is S(VI) rather than S(IV), and we use a sulfur fuel content of 600 ppm. The ICAO engine databank provides a hydrocarbon emissions index, which we multiply by 1.16 to obtain the total organic gas (TOG) emissions index. The engine databank also provides a total NO_x emissions index, which we speciate as 76% NO, 23% NO_2, and 1% HONO (Wood et al., 2008).

To be represented in the model, TOG must be further speciated into its component parts. We used a breakdown of TOG components from a joint FAA/EPA document describing best practices for quantifying organic gas emissions from aircraft engines (FAA and EPA, 2009). This document is based upon a series of recent aircraft measurement campaigns and is a substantial update to the previous approach developed by Spicer et al. (1994), which has been used for speciating TOG to-date. The new approach divided TOD into 77 explicit organic compounds and 4 categories of unknown compounds. The AMSTERDAM model, which uses a modified version of the Carbon Bond IV chemical mechanism, cannot represent each of these compounds separately. Rather, it requires inputs to be grouped according to their chemical properties. Therefore, the organic compounds needed to be "mapped" to species AMSTERDAM understands. Ratios provided by the U.S. EPA were used to create mass fractions for the lumped chemical species required by AMSTERDAM's chemical mechanism. The mapping process reduced the 77 compounds and 4 categories of unknown compounds to 8

compounds represented by AMSTERDAM: PAR, OLE, TOL, XYL, FORM, ALD2, ETH, and ISPD.

All of these emissions calculations were performed for the 17 most commonly used engines at Hartsfield-Jackson. This encompassed 96.5% of the total aircraft activity at the airport. The remaining 3.5% of activity was represented as a weighted average of the top 17 engines. Via this process, we obtained total annual emissions values for CO, NO, NO_2, HONO, H_2SO_4, SO_2, organic PM, elemental carbon, and eight organic gases in each of the four flight modes.

We made the simplifying assumption that the relative frequency of use of different engine models was constant throughout the year, so the quantity of emissions in each one-hour timestep was directly proportional to total aircraft activity in that timestep. Information on the relative frequency of flights in each month of the year, day of the week, and hour of the day at Atlanta International Airport was based upon previously generated emission inventories for Atlanta used by Arunachalam et al. (2008, 2011). We used these activity ratios and the total annual emissions for each flight mode to calculate timestep-specific emissions for each flight mode. Each mode's emissions were divided evenly among the emitters we defined for that mode in every timestep.

5B) Placing Emitters to Represent Air Traffic

In AMSTERDAM, the location of each emitter in 3D space is defined in a single, time-invariant data file. The number and locations of emitters cannot be changed over the course of a model run. Therefore, it was necessary to determine a static arrangement of emitters that would best represent the totality of aviation emissions from the airport.

Hartsfield-Jackson Atlanta Airport has five runways, all oriented in the East-West direction and ranging from 2740 – 3624 meters in length (Figure 1). However, in the model year (2002), the southernmost runway was not yet operational. Of the remaining four runways, the outer two are used for arriving planes, while the inner two are used for departures. At any given time, landings and take offs are either conducted from West to East or from East to West depending on the prevailing winds. A study analyzing flight operations in 1998-1999 found that the two operation directions were used with approximately equal frequency (Gladstone, 2000). We assume both directions of operation were equally frequent during our modeling period, so emissions are split evenly between the two directions.

Figure 1 - Hartsfield-Jackson Atlanta International Airport with a subset of emitters as represented in Google Earth. Yellow = taxi mode emitter, Pink = take off mode emitter, Blue = climb out mode emitter, Green = approach mode emitter. Red "E" and "W" markers are reference points at the ends of the runways relative to which emitters are placed.

A total of 51 emitters were used in the model. These emitters were divided into four modes: take off, climb out, approach, and taxi. For each departure runway, two emitters were used to produce take off mode emissions and five emitters were used to produce climb out mode emissions in each direction, for a total of 28 emitters. For each

arrival runway, five emitters were used to produce approach emissions in each direction, for a total of 20 emitters. Three emitters were used to produce taxi emissions: one positioned near the taxiways for the upper pair of runways, one near the taxiways for the lower pair of runways, and one near the gates.

Within each flight mode, each emitter receives an equal share of that mode's emissions in each timestep, so we chose to place emitters so as to most evenly cover the space in which aircraft operate in each mode, taking into account the fact that accelerating or decelerating aircraft spend more time in one part of that space than another. Aircraft were assumed to have constant acceleration and deceleration during each mode of flight, starting and ending at the speeds shown in Table 1. These speeds were used in conjunction with ICAO's official time-in-mode estimates to determine the linear distance traveled by aircraft in each mode (ICAO, 1993). Aircraft were assumed not to turn or change heading within the lowest 3000 ft of altitude, the region included in this study. The distance covered by aircraft in each flight mode was divided into segments that represent equal time intervals rather than equal lengths (so that mode-specific emissions would be equal in each segment). An emitter was positioned at the time-based center of each segment (i.e., the spot where a plane would be located when half of the time that plane spent in a segment had elapsed). The path of travel in each mode of flight was assumed to be linear, except for the takeoff mode, wherein the first segment was assumed to be along the runway and the entire altitude gain was included in the second segment. The approach mode was assumed to terminate at the point where aircraft first touch the runway. The take off mode was assumed to begin near the end of

each runway, and the climb out mode begins at the point in the air where the take off mode ends.

Mode	Speed at Start of Mode	Speed at End of Mode
Take Off	0 ft/s	225 ft/s
Climb Out	225 ft/s	422 ft/s
Approach	422 ft/s	250 ft/s

Table 1 - Speeds at the start and end of flight modes. Speeds estimated based on discussion with the FAA and aircraft statistics from Air New Zealand (2011)

Using these assumptions, the horizontal and vertical position of each emitter was calculated. Google Earth was used to position emitters relative to the airport runways and to obtain latitude and longitude coordinates for each emitter. Ultimately, climb out emitters spanned six grid cells (24 km) in the East – West direction and approach emitters spanned twelve grid cells (48 km) in the East – West direction. Vertically, emitters spanned the bottom 11 layers of the 19-layer domain.

5C) Comparison of Aviation Emissions by Chemical

Table 2 compares the total annual emissions of several chemicals used in TC1 (the base case), in TC4 (the ADSC case), and the emissions totals below 3000 feet produced by EDMS, which was used in past work such as that by Arunachalam et al. (2008, 2011). The emissions used in this study were less than the emissions calculated by EDMS for all species except NO_2.

Note that Arunachalam et al. (2011) used a research version of EDMS 5.0 in which the $PM_{2.5}$ speciation was based upon FOA version 3a (CSSI, 2007). Also note that while the NO_x speciation in this study was based on Wood et al. (2008), Arunachalam et

al. (2008, 2011) used a speciation profile based on the London Heathrow Airport study (UK Dept for Transport, 2006).

Species	TC1	TC4	EDMS	TC4/TC1	EDMS/TC1
CO	3,923,571	3,923,571	4,963,366	100%	127%
NO	3,395,707	3,395,707	3,664,692	100%	108%
NO_2	1,027,648	1,027,648	558,514	100%	54%
Sulfate PM	14,198	14,055	39,456	99%	278%
SO_2	454,324	454,324	515,342	100%	113%
Org PM	3,318	3,318	19,271	100%	581%
EC	11,340	61,557	14,931	543%	132%
TOG	351,081	351,081	1,101,578	100%	314%

Table 2 - Comparison of total annual aircraft emissions (in kg) from the Atlanta airport at altitudes below 3,000 ft for TC1, TC4, and EDMS-derived emissions used in Arunachalam et al., (2008, 2011). The last two columns contain ratios of total emissions used in different test cases or studies.

6. Results and Analysis

6A) Model Performance

Model performance was evaluated by comparing results of all four test cases to measurements from the Chemical Speciation Network (CSN). Within our modeling domain, there were 11 CSN monitors. Results were compared both with all of these CSN monitors (averaged) as well as the single monitor closest to the airport, the Decatur monitor [130890002]. This monitor is located to the northeast of the airport, roughly 6-7 miles from the airport's edge.

Table 3 compares our model performance across the 11 CSN monitors within our domain to our model performance at the single monitor closest to the airport (the Decatur monitor). The observation recorded by each monitor at each time period is compared to

the model's predicted concentration at that same time period. These results are averaged over time to produce mean observed values and mean modeled values, and the normalized error and bias of the model are calculated. Results are presented for both the Decatur monitor and the average of all 11 CSN monitors.

The 11 CSN monitors are widely spread across Georgia and Alabama. Since aircraft only contribute a small fraction of total emissions, model performance is overwhelmingly determined by non-aviation factors (such as the background emissions, meteorology, etc.) in the case of the CSN 11-monitor average. The results show that the model has smaller error and bias for the Decatur monitor alone in the case of every pollutant except NH_4 and SO_4. NH_4 has lower error but slightly more bias, while SO4 has increased error and bias. In general, the results for the Decatur monitor appear to be better than those for the 11-monitor average. This may be because the Decatur monitor is near the center of the model domain, and therefore it may be less significantly affected by the model's boundary conditions (concentration and meteorology) than the other CSN monitors.

Overall, our model results appear to be accurate within an order of magnitude. Looking at the Decatur monitor, model results were approximately accurate for elemental carbon, were biased 6% high for nitrate, and were biased 30% - 80% low for the other five pollutants.

A low bias for non-reactive species such as primary organic aerosol (the most important component of organic PM in our simulation, comprising 82% of aircraft impact on total organic PM concentration in the airport's grid cell) could have the effect of

increasing the apparent aircraft contribution to organic PM. This is because an artificially low background concentration of organic PM will reduce the coagulation rate of POA particles. Since larger particles deposit out of the atmosphere more quickly, a slower coagulation rate will reduce deposition of organic PM from aircraft emissions, increasing the apparent contribution of aircraft to organic PM.

A low bias for a secondary species (formed via reactions in the atmosphere) such as sulfate may have the opposite effect, reducing the apparent aircraft contribution to that species. A lower background concentration of a secondary chemical likely indicates a lower concentration of the reactants which form that chemical. The reactants emitted by aircraft will react more slowly with an artificially low background concentration of reactants, forming less of the secondary species.

		CSN Avg.	Decatur Monitor
EC	test case mean	0.463	0.762
	observed mean	0.549	0.763
	norm mean error	44.8%	42.2%
	norm mean bias	-15.6%	-0.2%
NH4	test case mean	1.085	0.970
	observed mean	1.500	1.367
	norm mean error	41.8%	40.6%
	norm mean bias	-27.7%	-29.1%
NO3	test case mean	0.341	0.464
	observed mean	0.537	0.436
	norm mean error	78.7%	72.3%
	norm mean bias	-36.6%	6.3%
OC	test case mean	0.856	1.024
	observed mean	4.695	4.553
	norm mean error	81.8%	77.5%
	norm mean bias	-81.8%	-77.5%
PM25	test case mean	8.211	8.507
	observed mean	18.621	18.418
	norm mean error	55.9%	53.8%
	norm mean bias	-55.9%	-53.8%
SO4	test case mean	3.065	2.425
	observed mean	5.846	5.696
	norm mean error	48.8%	58.8%
	norm mean bias	-47.6%	-57.4%
TC	test case mean	1.322	1.787
	observed mean	5.252	5.316
	norm mean error	74.8%	66.4%
	norm mean bias	-74.8%	-66.4%

Table 3 - Comparison of TC1 Model Results (in μg/m^3) to monitor observations from the 11 CSN monitors and from the Decatur monitor in June and July 2002. Normalized mean error and normalized mean bias are shown.

Table 4 compares the mean value in each of our model runs to the mean value at the Decatur monitor. Normalized mean error and normalized mean bias are also shown. All four test cases produced mean results that match to within a few hundredths of a μg/m^3, and normalized mean error and bias typically match to within 1%. For most chemicals, this may exceed the sensitivity of the measurements from the monitoring

networks, considering that uncertainty is introduced into measurements by volatility of some compounds, calibration issues, and limitations of available analytical methods (Eder, 2005). Notably, the differences between TC1 (the base case) and TC3 (the case without the PinG process) are too small to draw meaningful conclusions about whether the PinG process increases or decreases model accuracy or precision.

The largest difference between test cases occurs when comparing results for elemental carbon between TC1 (the base case) and TC4 (the ADSC case), which had 543% more elemental carbon from aviation emissions than TC1. This caused the bias to increase from -0.2% to 2.1%. As shown in Table 2, our calculations regarding elemental carbon (used in TC1) are similar to those provided by EDMS and used in past studies; the measurement-based emissions index used in the ADSC model appears to be the outlier. The model performance analysis implies that the measurement-based elemental carbon emissions index may be less accurate than the calculated index used in TC1 for this application.

However, since aircraft are a small contributor to total concentrations, it is also possible that accurately modeling aircraft contribution can decrease model precision, depending on the original bias of the concentrations resulting from background emissions sources alone. We see this effect most clearly by comparing nitrate concentrations for TC1 and TC2 (the case without aircraft). TC1 necessarily models aircraft more accurately than TC2, which omits them entirely. Nevertheless, TC2 has less error and less bias because the model was biased high for nitrate, so omitting aircraft results in concentrations that are more accurate. Thus, one cannot assume that a change in the

methodology used to model aircraft must be an improvement if it improves model accuracy and bias.

		TC1	TC2	TC3	TC4
EC	test case mean	0.762	0.759	0.762	0.779
	observed mean	0.763	0.763	0.763	0.763
	norm mean error	42.2%	42.1%	42.2%	42.8%
	norm mean bias	-0.2%	-0.6%	-0.2%	2.1%
NH4	test case mean	0.970	0.965	0.967	0.970
	observed mean	1.367	1.367	1.367	1.367
	norm mean error	40.6%	40.7%	40.6%	40.6%
	norm mean bias	-29.1%	-29.4%	-29.3%	-29.1%
NO3	test case mean	0.464	0.460	0.461	0.464
	observed mean	0.436	0.436	0.436	0.436
	norm mean error	72.3%	71.9%	72.0%	72.2%
	norm mean bias	6.3%	5.4%	5.6%	6.3%
OC	test case mean	1.024	1.023	1.024	1.025
	observed mean	4.553	4.553	4.553	4.553
	norm mean error	77.5%	77.5%	77.5%	77.5%
	norm mean bias	-77.5%	-77.5%	-77.5%	-77.5%
PM25	test case mean	8.507	8.482	8.494	8.524
	observed mean	18.418	18.418	18.418	18.418
	norm mean error	53.8%	53.9%	53.9%	53.7%
	norm mean bias	-53.8%	-53.9%	-53.9%	-53.7%
SO4	test case mean	2.425	2.414	2.420	2.426
	observed mean	5.696	5.696	5.696	5.696
	norm mean error	58.8%	59.0%	58.9%	58.8%
	norm mean bias	-57.4%	-57.6%	-57.5%	-57.4%
TC	test case mean	1.787	1.782	1.786	1.804
	observed mean	5.316	5.316	5.316	5.316
	norm mean error	66.4%	66.5%	66.4%	66.1%
	norm mean bias	-66.4%	-66.5%	-66.4%	-66.1%

Table 4 - Comparison of model results (in μg/m^3) for four test cases to observations from the Decatur monitor in June and July 2002. Normalized mean error and normalized mean bias are shown.

6B) Aircraft Impacts

For the base testcase (TC1), the model predicted average $PM_{2.5}$ concentrations from 10 - 12.2 μg/m^3 in urban centers, with concentrations as low as 4.7 μg/m^3 in rural regions (Figure 2). The airport does not stand out in this figure because aircraft

contribution to average surface $PM_{2.5}$ was only 0.23 $\mu g/m^3$ in the airport's grid cell. This graph is included primarily to provide context for the upcoming graphs. Note that this is the only graph in this paper that shows the results of a single test case rather than the difference between a pair of test cases. Accordingly, this graph has units of $\mu g/m^3$ and uses its own color scale. All subsequent graphs have units of ng/m^3 and share a single color scale.

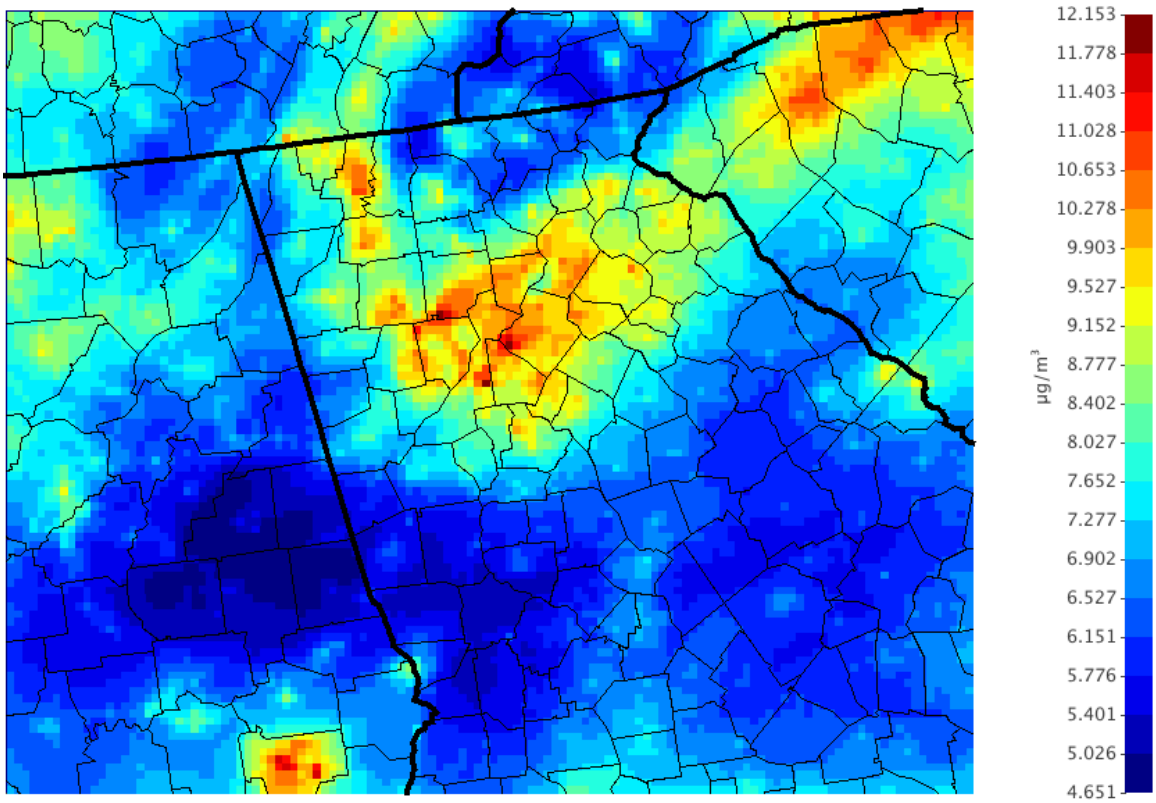

Figure 2 - Average modeled surface $PM_{2.5}$ concentrations ($\mu g/m^3$) in June and July 2002 in TC1 (base case)

Aircraft increased average concentration of total $PM_{2.5}$ in June and July, 2002 near the airport and throughout the Atlanta metro area (Figure 3). Concentrations in the grid cell containing the airport were elevated by 237 ng/m^3. In the airport's grid cell, the species that added or subtracted at least 1% to aircraft impact on $PM_{2.5}$ were: sulfate

(58%), ammonia (18%), elemental carbon (16%), primary organic aerosol (9%), and nitrate (-2%). Total $PM_{2.5}$ concentrations in some outlying areas were actually slightly reduced by aircraft due to a reduction in sulfate concentration.

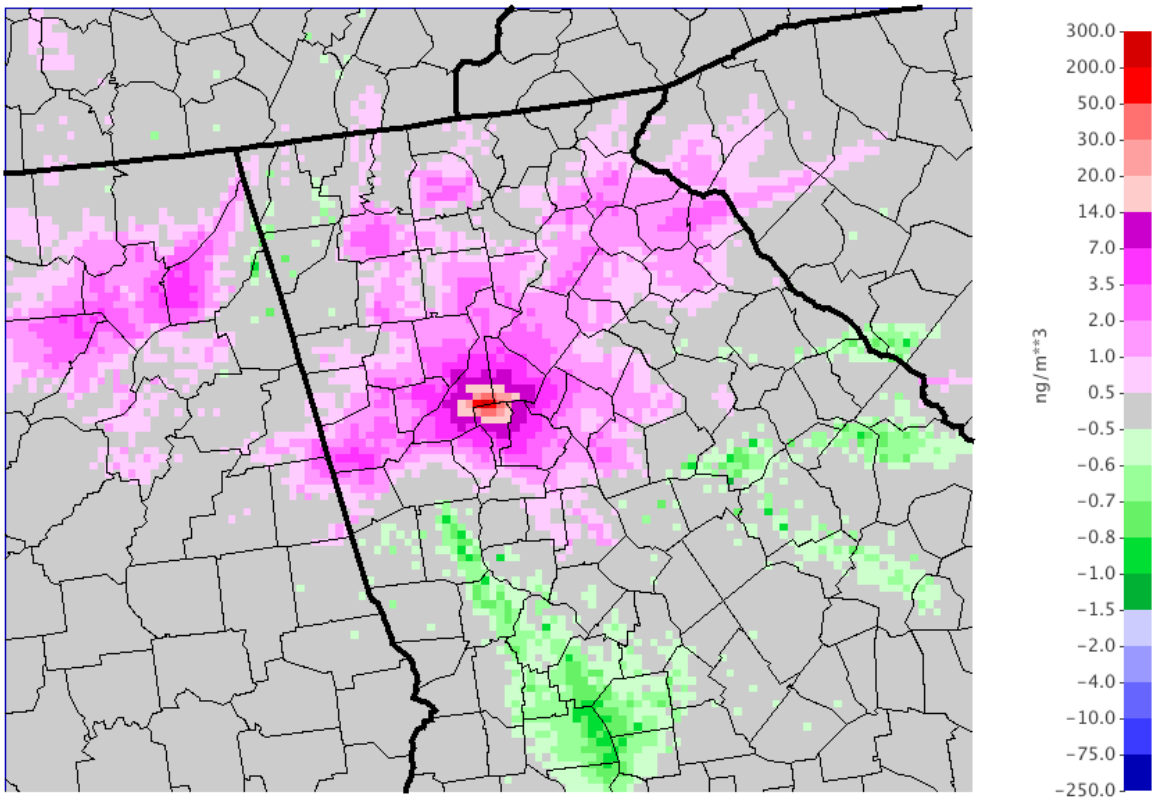

Figure 3 - Aircraft impact on average surface $PM_{2.5}$ concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no aircraft case TC2)

Near the airport, aircraft raised sulfate concentrations by up to 136 ng/m^3 (Figure 4). In some areas far from Hartsfield-Jackson, aircraft caused small reductions in sulfate aerosol concentrations (<1 ng/m^3). Tsai et al. observed a similar result when using a Lagrangian plume model to investigate aircraft impacts on sulfate pollution over Taiwan (2001). They attributed the reduction to the fact that aircraft plumes are rich in SO_2 emissions. The SO_2 initially reacts with OH radicals to form sulfate aerosol. However, in aged plumes, OH radicals become depleted, slowing sulfate formation. In our

simulation, there must be sufficient sulfate removal processes to lower sulfate aerosol concentration in aged plumes below the sulfate concentration of ambient air.

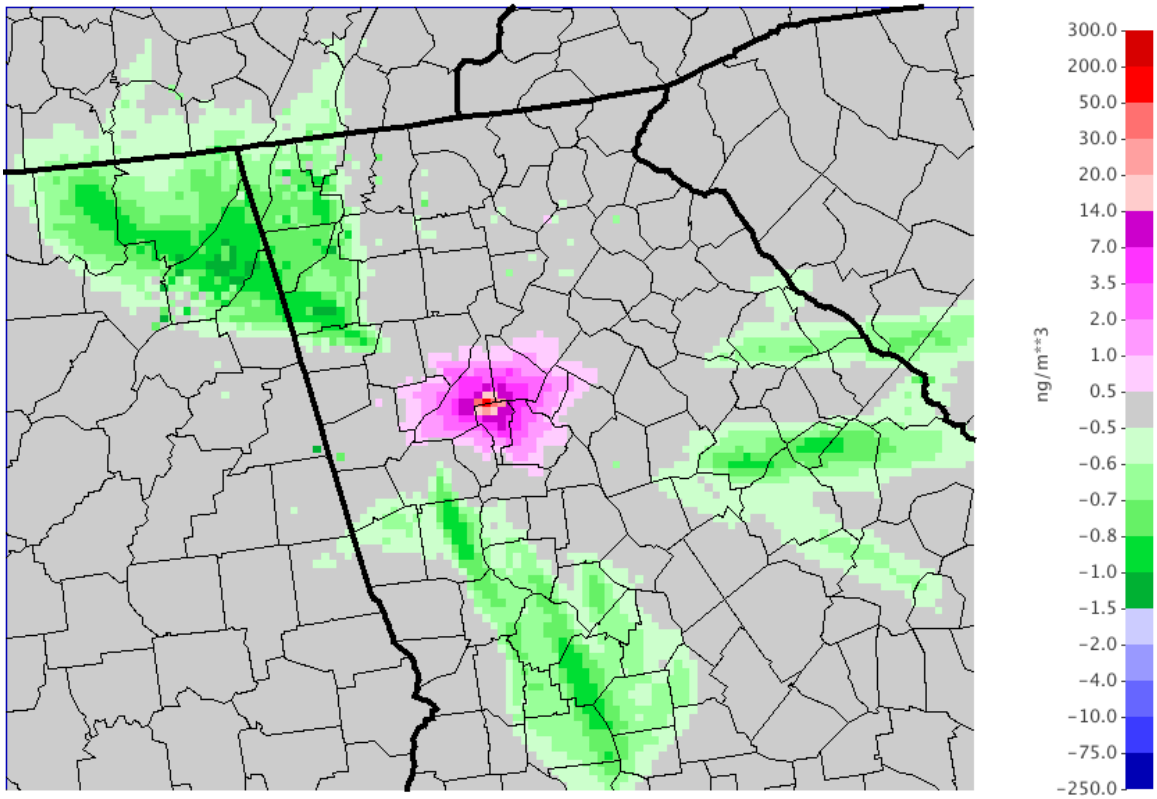

Figure 4 - Aircraft impact on average surface sulfate aerosol concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no aircraft case TC2)

Aircraft increase elemental carbon concentrations near the airport by 38 ng/m^3 (Figure 5). Elemental carbon impacts taper off quickly with distance from the airport. Elemental carbon is non-reactive in the AMSTERDAM model. Therefore, the increase is due to the addition of direct elemental carbon emissions and the effect tapers with distance due to dilution (i.e., more volume is available in rings at greater radii from the airport) and deposition.

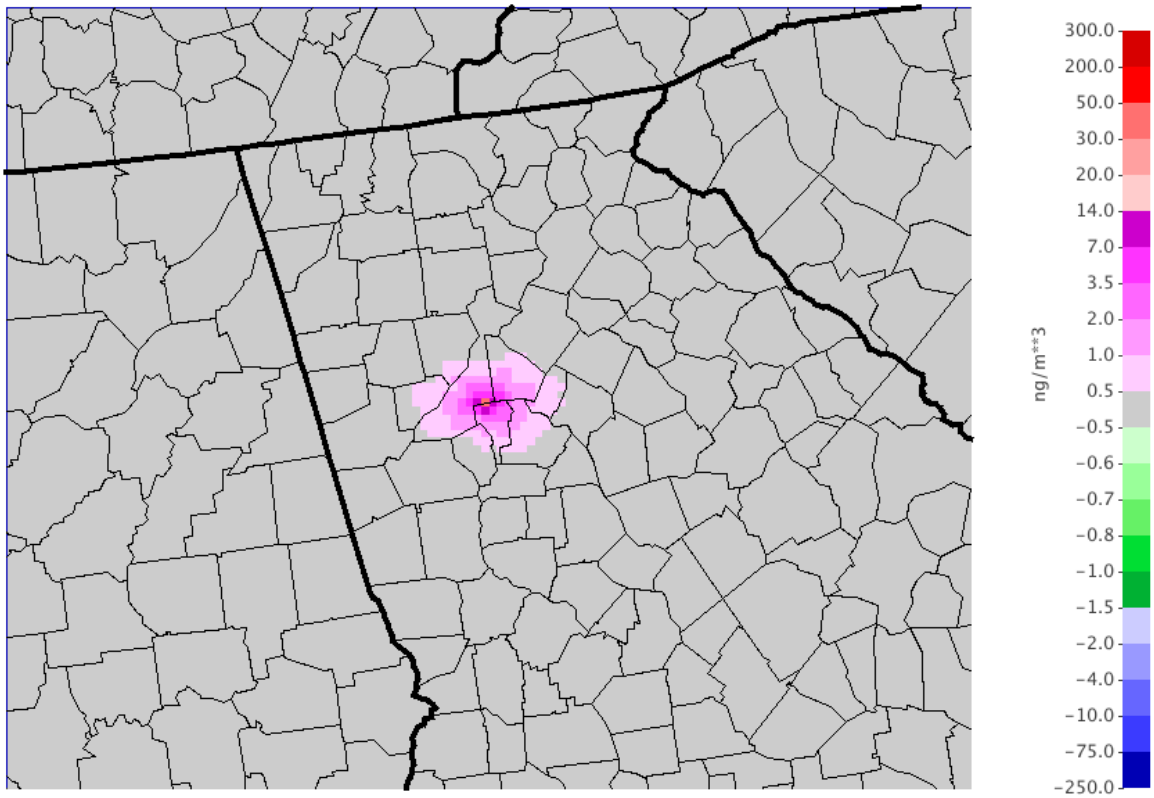

Figure 5 - Aircraft impact on average elemental carbon concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no aircraft case TC2)

Aircraft have a small positive effect on organic PM concentrations (Figure 6). Almost all of this aircraft contribution was due to primary organic aerosol (POA). Like elemental carbon, POA is treated as a non-reactive species in CMAQ, so again the decrease with distance from the airport is due to both deposition and dilution at greater radii from the emissions source.

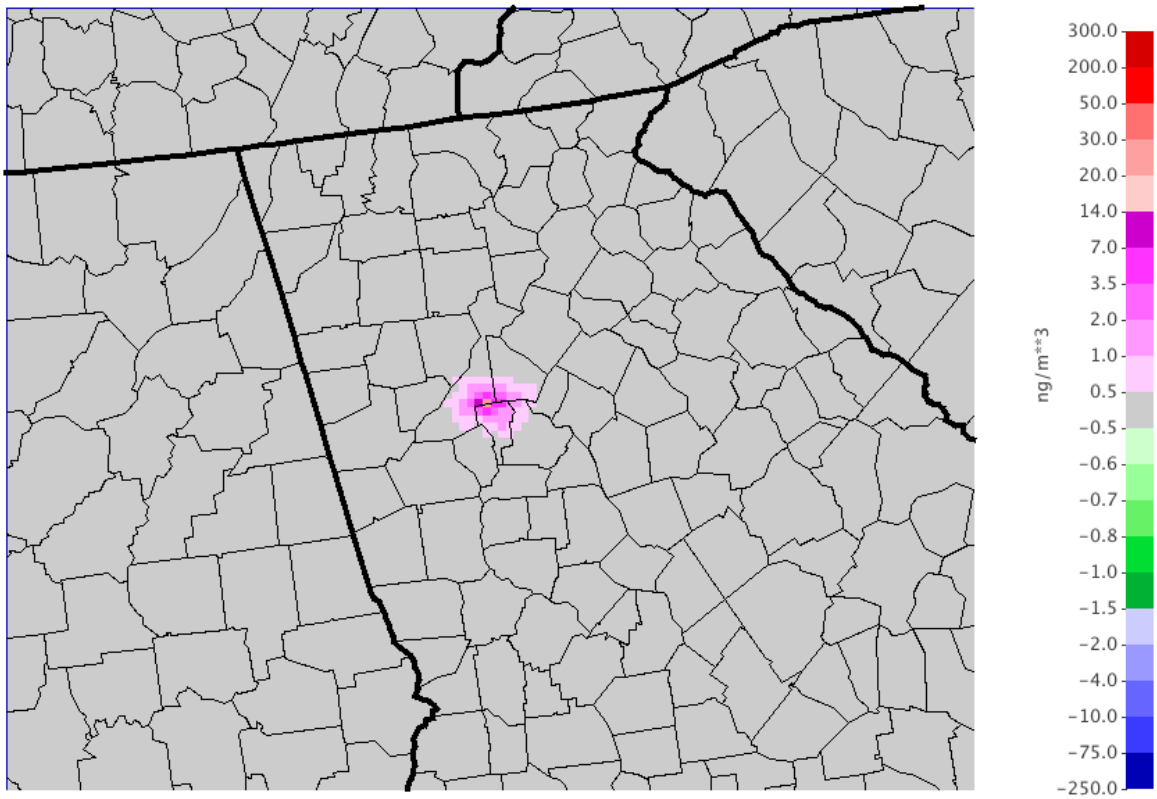

Figure 6 - Aircraft impact on average organic PM concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no aircraft case TC2)

6C) Effects of the Plume-in-Grid Process

Figures 7 – 11 show the result when TC3 (Regular CMAQ test case) is subtracted from the TC1 (the base case, with PinG process). Hence, positive values indicate that the plume-in-grid process resulted in higher concentrations, while negative values indicate the regular CMAQ simulation resulted in higher concentrations.

In most grid cells, the use of the PinG process increased total $PM_{2.5}$ concentrations. This may be due to a higher concentration of reactants in the puffs, which are more likely to collide with each other when they are not diluted in a 4 km grid cell. Similarly, accumulation processes, which rely on collisions between particles of the same species, should proceed faster when the same quantity of emissions is confined to a smaller volume.

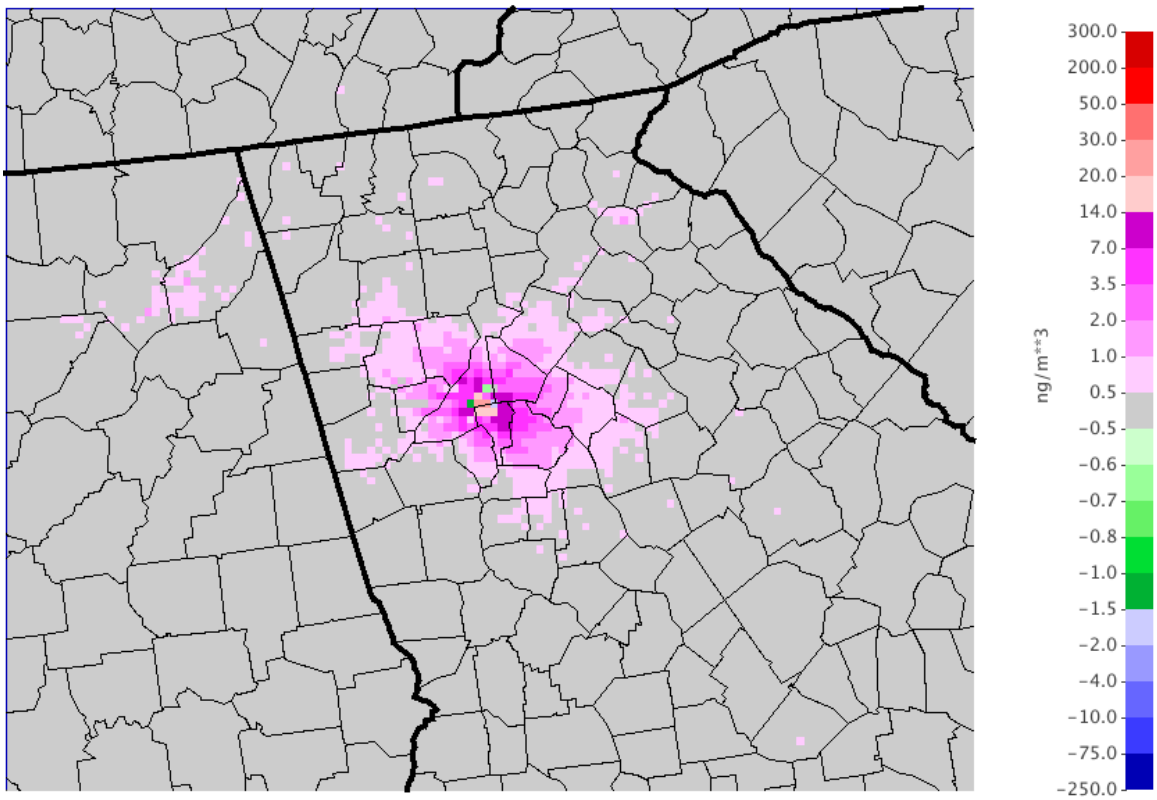

Figure 7 - Effects of plume-in-grid process on average $PM_{2.5}$ concentrations (ng/m3) in June and July 2002 (i.e., base case TC1 minus no PinG case TC3)

Note that except in the two cells nearest the airport, the entire East-West line of cells containing the aircraft approach and departure paths exhibited a lower increase due to the PinG process than rows immediately above and below (this row is outlined in Figure 8), indicating that there is a countervailing phenomenon operating these cells, but the magnitude of this phenomenon usually is not large enough to outweigh the processes which increase concentrations with PinG emitter usage across most of the map. It is possible that the high altitude of emitters in these cells causes a larger-than-normal amount of PM to remain above the ground layer in TC1, whereas downward movement of pollutants may be more rapid in TC3 when pollutants are not confined to puffs. One possible explanation for this difference is that puffs are added to the model at a higher temperature than the surrounding air and experience heat-driven "plume rise" which

carries them upward. Pollutants added directly to the 4 km grid cell do not experience heat-driven plume rise.

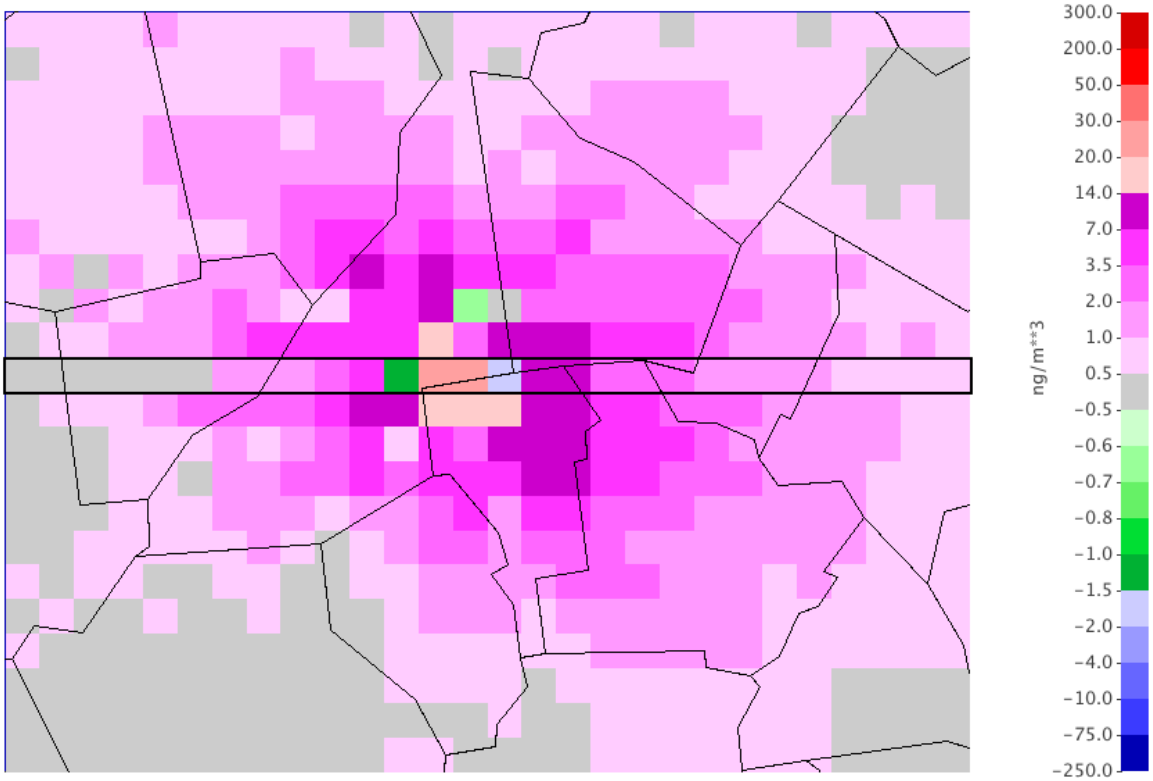

Figure 8 - Effects of plume-in-grid process on average $PM_{2.5}$ concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no PinG case TC3), zoomed in on the airport. The row containing most take off, climb out, and approach emitters is outlined. Away from the airport, this row exhibits a smaller increase in aircraft impact due to the PinG process than rows above and below, and two cells in this row exhibit a decrease in aircraft impact due to the PinG process.

The effects of the plume-in-grid process for sulfate aerosol (Figure 9) exhibited a similar pattern to those for overall $PM_{2.5}$. Use of the PinG process increased concentrations everywhere, especially at the airport itself. As before, increases were lower along the approach and departure paths where above ground emitters were situated, possibly indicating that more of the $PM_{2.5}$ remained above ground level in those areas in the PinG simulation.

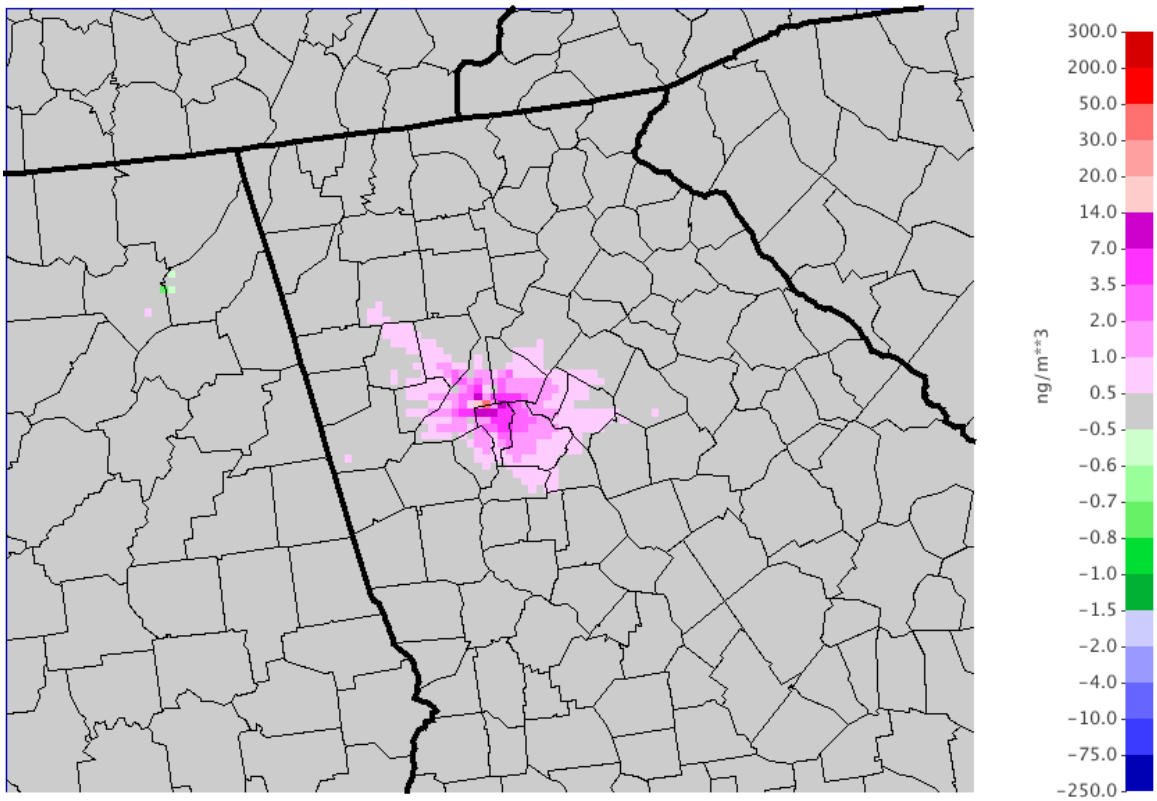

Figure 9 - Effects of plume-in-grid process on average sulfate aerosol concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no PinG process case TC3)

In most areas, particularly at the airport and along the approach and departure flight paths, use of the PinG process significantly lowered elemental carbon concentrations (Figure 10). This is in contrast to results for sulfate, where the PinG process generally increased concentrations (Figure 9).

One possible explanation for this difference stems from the fact that elemental carbon is non-reactive while sulfate can be formed in the atmosphere. By concentrating sulfate precursors, the PinG process enables sulfate aerosol to be generated more rapidly. In contrast, elemental carbon particles are emitted directly and concentrating them does not cause more of them to come into existence. However, increasing EC concentration can increase the total amount of elemental carbon that is deposited out of the atmosphere, because accumulation processes will be faster in a smaller volume, and larger particles

deposit more quickly than smaller ones. (While sulfate would also be deposited more rapidly in the PinG scenario, this is outweighed by the faster rate of formation.)

There are some areas near the airport where the PinG process increases elemental carbon concentrations. Chemicals may be transferred between grid cells in a somewhat different pattern when the chemicals are confined to puffs rather than spread throughout grid cells. Based on the usual winds and physical properties near the airport, this may lead to slightly higher values in some grid cells and slightly lower values in others.

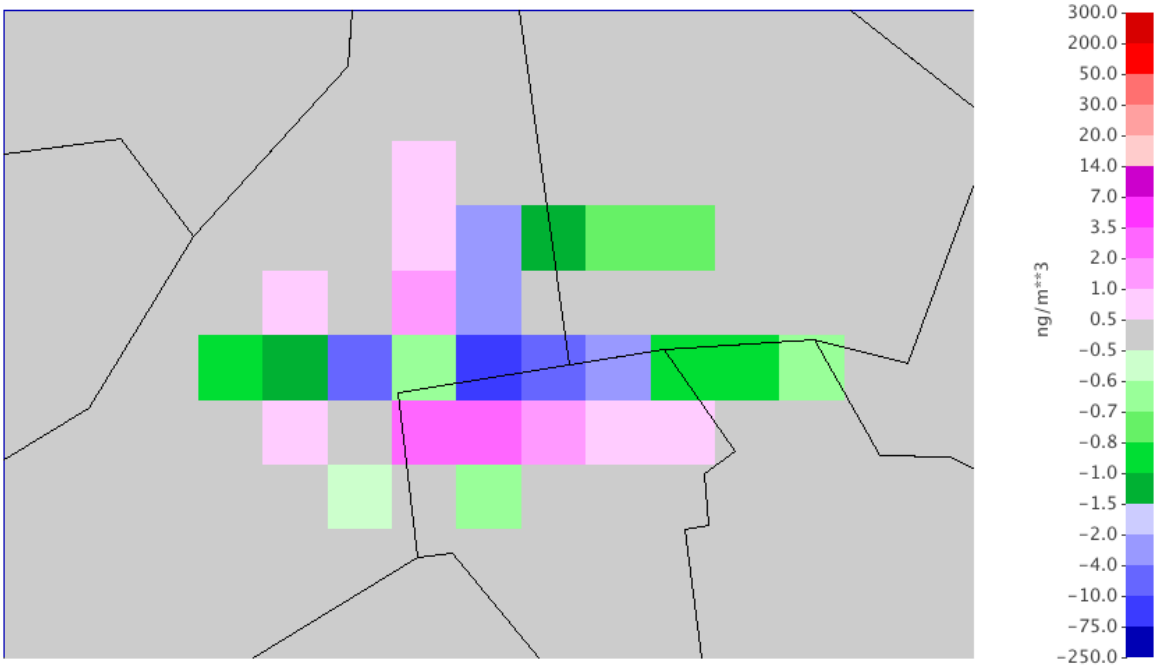

Figure 10 - Effects of plume-in-grid process on average elemental carbon concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no PinG process case TC3), zoomed in on the airport. No effects exceeding ±0.5 ng/m^3 were observed outside of the domain depicted.

The results for organic PM are mixed (Figure 11). In most of the grid cells where emitters are located, the PinG process reduced organic PM concentrations. Aircraft influence on organic PM is overwhelmingly due to POA, which is considered to be non-reactive in AMSTERDAM and therefore may undergo increased deposition for the same reasons as outlined for elemental carbon above.

However, a considerable number of cells in Figure 11, including one in the aircraft approach path, exhibit increased concentrations due to the PinG process. This may be due to differences in chemical movement between cells when chemical are confined to puffs, as in the case of elemental carbon.

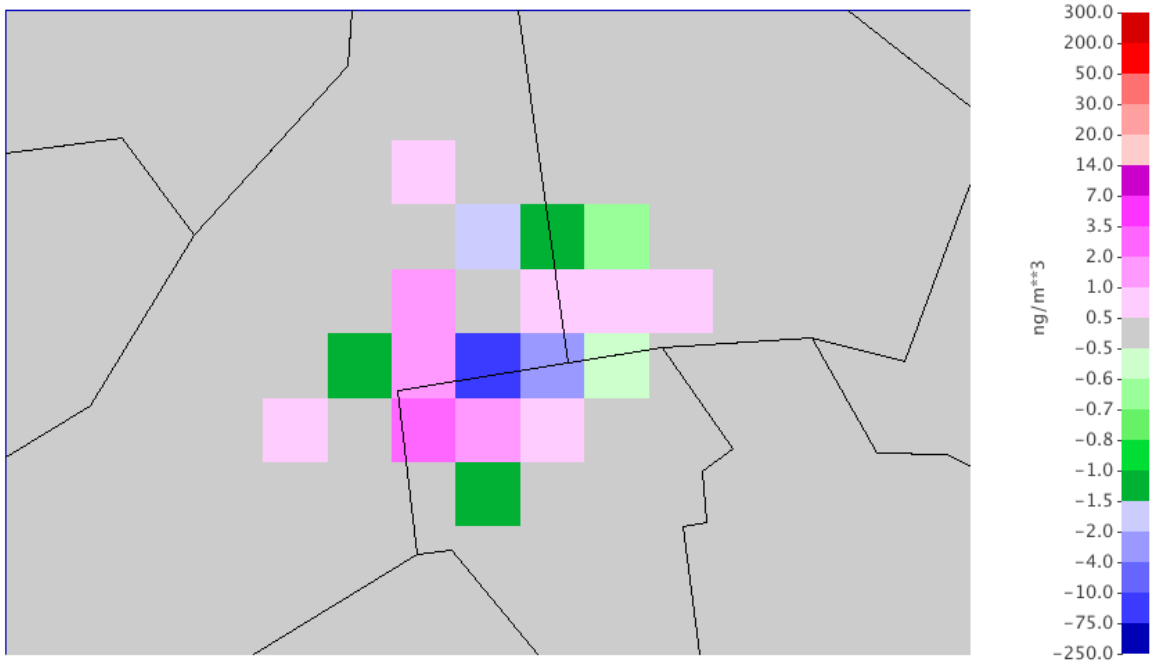

Figure 11 - Effects of plume-in-grid process on average organic PM concentrations (ng/m^3) in June and July 2002 (i.e., base case TC1 minus no PinG process case TC3). No effects exceeding ±0.5 ng/m^3 were observed outside of the domain depicted.

6D) ADSC Alternative Emissions Evaluation

In TC4 (the ADSC test case), sulfate emissions were 99% of TC1 while elemental carbon emissions were 543% of TC1. Figures 12 and 13 show the ADSC case subtracted from the base case, so negative values indicate where the ADSC case is higher. Sulfate differences were small, ranging from 1.6 ng/m^3 higher to 0.7 ng/m^3 lower in TC1 than in TC3 (Figure 12).

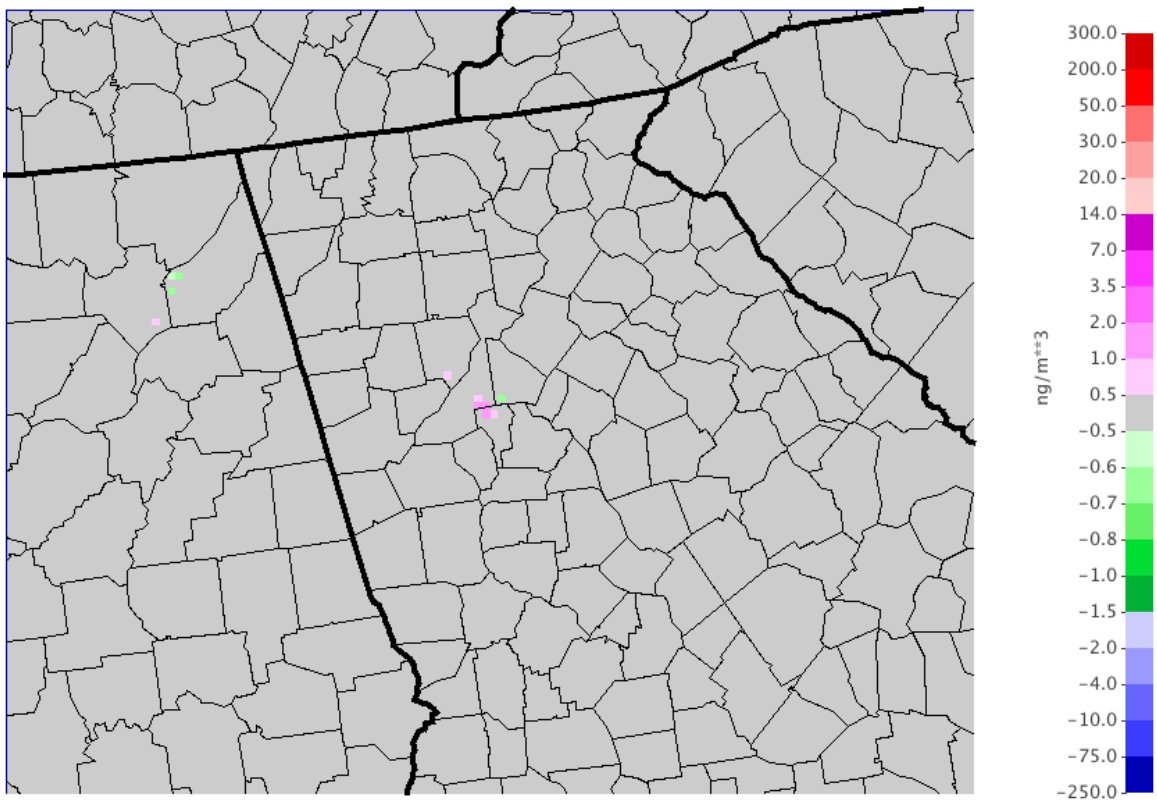

Figure 12 - Base case (TC1) minus ADSC case (TC4, with 99% of the sulfate aircraft emissions of TC1) for average sulfate (ng/m^3) in June and July, 2002

The ADSC test case featured substantially higher elemental carbon concentrations near the airport and throughout the Atlanta region. This is because elemental carbon emissions were much higher in the ADSC test case. Elemental carbon is relatively nonreactive in the atmosphere, so there are few factors that would influence concentrations other than emission and deposition rates.

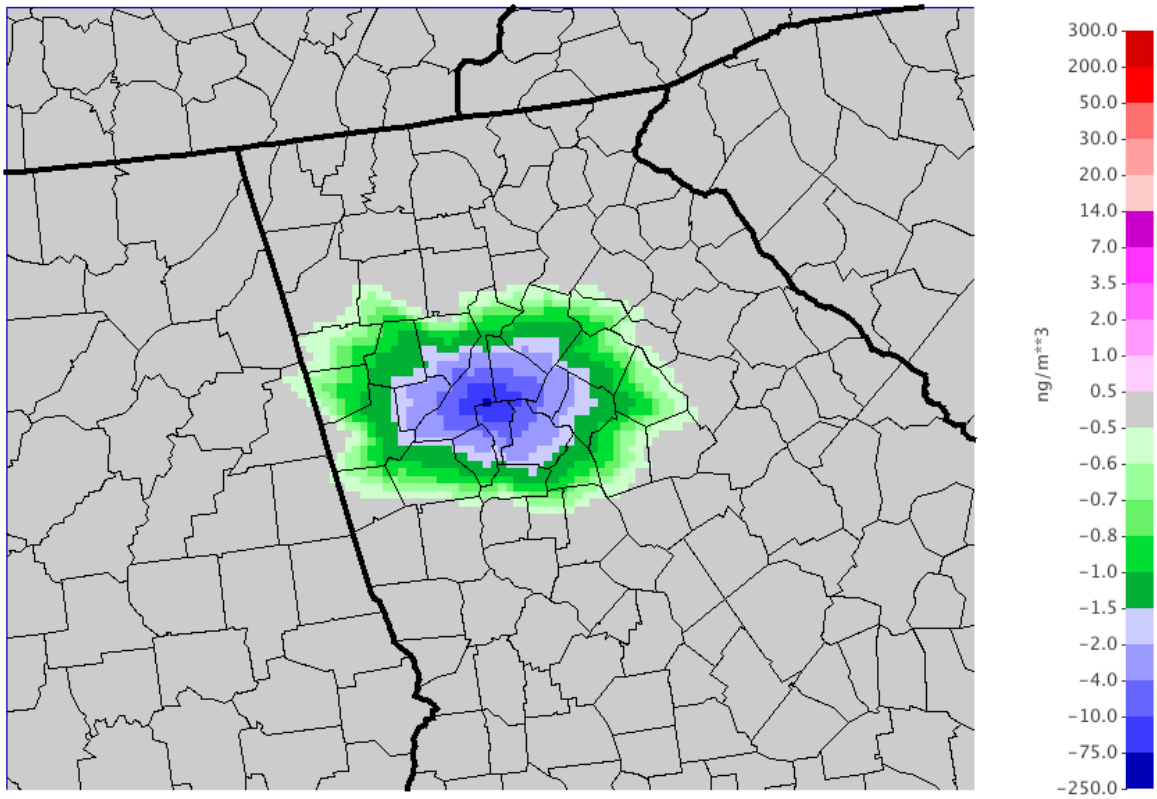

Figure 13 - Base case (TC1) minus ADSC case (TC4, with 543% of the elemental carbon aircraft emissions of TC1) for average elemental carbon (ng/m^3) in June and July 2002

7. Conclusions

In June and July, 2002, aircraft contributed 237 ng/m^3 to average $PM_{2.5}$ concentrations in the airport's grid cell and between 1 and 7 ng/m^3 in a multi-county region covering thousands of square kilometers. At the airport, aircraft contribute 58% of total $PM_{2.5}$ as sulfate aerosol, 18% as ammonia, 16% as elemental carbon, and 9% as primary organic aerosol. They reduce nitrate concentrations by 2%. Farther from the airport, aircraft decrease sulfate concentrations by up to 1 ng/m^3 due to depletion of OH radicals required for sulfate formation.

Use of a plume-in-grid process for modeling air pollutants tends to increase the concentrations of secondary PM pollutants by 0.5 – 20 ng/m^3 in a large area surrounding the airport. For non-reactive primary pollutants, such as POA and EC, the PinG process

tends to lower concentrations in grid cells where emitters are located (by as much as 13 ng/m^3), possibly due to increased coagulation and resulting deposition. However, concentrations are sometimes increased in nearby areas by up to 3 ng/m^3, likely due to differences in pollutant transport when pollutants are confined to puffs.

It is unclear if the plume-in-grid process improves model performance for this analysis because the model performance analysis did not find significant differences in error and bias between TC1 and TC3. Use of the plume-in-grid process on a larger and more isolated source, such as a rural power plant, may be more likely to produce significant differences in error and bias relative to monitor values. Since the plume-in-grid process tends to increase $PM_{2.5}$ concentrations within about 50 km of modeled sources (Figure 7), a researcher whose model results are too low in proximity to sources of interest might consider using a plume-in-grid process as a means of rectifying the problem.

At the 4 km grid cell resolution, for many applications, differences in modeled concentrations may not justify the extra effort required to use a plume-in-grid process. However, if results were to be examined on a sub-grid scale, it may be possible to derive additional value from the plume-in-grid process not seen in these results.

Increasing the emissions of elemental carbon by 543% (for the ADSC test case) resulted in higher modeled elemental carbon concentrations (212 ng/m^3 at the airport, tapering off with distance). Model performance analysis indicated that this alteration increased error by 0.6% and changed bias from -0.2% to 2.1% for elemental carbon.

8. Future Work

There are a number of projects future researchers might consider to extend this work.

Many airports around the country have existing emissions inventories, which were produced via the EDMS model. A program could be developed which reads gridded EDMS emission outputs and generates a set of PinG input files for AMSTERDAM on the basis of those emissions (for instance, by placing an emitter in the center of each EDMS grid cell). This would allow AMSTERDAM to be used much more efficiently to model the impacts from many airports using existing EDMS-based emissions inventories.

A large amount of sub-grid scale data was generated for this project but was not able to be included in this analysis. Concentrations of pollutants in puffs may be substantially higher than concentrations in 4 km grid cells, resulting in increased estimates for peak exposure events. This may have important health consequences. Future researchers may wish to analyze the sub-grid scale data generated for this project, or generate new sub-grid scale data, to determine the maximum concentrations to which individuals may be exposed.

A future year scenario could be modeled, assuming changes in background emissions (due to new or improved control technologies, regulations, changes in GDP, etc.) and differences in aircraft emissions (due to changes in fleet composition, air travel demand, engine technology, etc.).

This work modeled the airport with a set of 51 emitters. Future researchers may consider using a greater number of emitters to determine if this has any impact on model performance and to determine the extent of increased computational resources necessary to perform a simulation with more emitters.

Appendix: Building and Fixing the AMSTERDAM Model

The uncompiled Fortran code for AMSTERDAM was provided to us by model developers Prakash Karamchandani and Krish Vijayaraghavan. The model contained numerous programming errors and bugs, some of which prevented the program from compiling or running on our system. Over nine months were spent analyzing the sparsely commented code, locating problems, and fixing bugs in the model.

Some of the bugs we identified may have affected previously published results involving the AMSTERDAM model (e.g., Karamchandani, 2010). If present, errors are most likely to be found in N_2O_5 output concentrations, mercury aerosol emissions, chemical contributions from puffs to the CMAQ grid, and chemistry in puffs when calculated by processors other than the main processor.

It is important to note that there are likely to be additional bugs in the AMSTERDAM model that we did not locate. Although we are unaware of any outstanding errors or problems in the model code or in our results, the likelihood of additional bugs should be considered a potential source of error in this work.

We have developed substantial materials on how to successfully build AMSTERDAM and correct or work around all of the bugs we identified. These materials are available in this appendix, for the benefit of other researchers interested in using the AMSTERDAM model.

Guide to Installing and Running the AMSTERDAM Model

Guide version 1.03, Feb 2011

Table of Contents

Introduction

This document is a guide to the process of installing and running the AMSTERDAM model on a linux cluster, such as the "Emerald" cluster at UNC-Chapel Hill. It covers installation of supporting libraries, configuration of AMSTERDAM, and resolutions for numerous bugs which remain in the model code provided by AER.

This document endeavors to provide the necessary details to build either of the included test cases (CMAQ_APT or MADRID), in single-processor or parallel processor configurations, with or without simulating Mercury, with the PGF or iFort compilers. All 16 combinations of possible builds have not been fully tested (in particular, testing on MADRID was very limited). However, many combinations have been tested, and this guide attempts to be as helpful as possible.

If you encounter errors in the build process, you should first check the "Error FAQ" near the end of this document. Your error, or a similar error, may have already been addressed.

This guide was developed by Jeffrey Rissman at the UNC Institute for the Environment.

A) Software and Hardware Versions Utilized

When developing this guide, the following software versions were utilized:
- PGF compiler 9.0.1
- iFort compiler 11.0
- mpich2 1.0.3
- netCDF 3.6.3
- IOAPI 3.0
- An AMSTERDAM build from AER packaged in 9 .tar files dated **8-27-09**

The guide was developed on machines with the following properties:
- 1.86 GHz Intel Xeon processors
- 64-bit x86 architecture
- 8 processors per machine, 4 cores per processor
- OS: Red Hat Enterprise Linux Server 5.4 (kernel version 2.6.18)

Part 1: Obtaining, Placing, and Extracting the Files

---

1. Log into your computer system. You likely start in your home folder. Make sure that you are using the c-shell (csh) or t-shell (tcsh), rather than the bourne-again shell (bash) or any other shell.

2. Make sure your package space is set up correctly. It should contain the Fortran compiler you are going to use followed by mpich2. It should not contain any other Fortran compilers. (I also find it necessary to remove "gcc" from my package space when building netCDF.)

   If you are using integrated package manager (IPM), you can check what items are in your package space by typing "ipm query" and remove one by typing "ipm remove [package_name]". When adding packages, you may wish to explicitly specify the version (e.g. "ipm add intel_fortran-110" not "ipm add intel_fortran") in case your IT organization installs a new version of this software and changes the unspecific link to point to the new version. If you manage your own computer cluster, this may not apply to you.

3. Move to the folder where you want to install CMAQ AMSTERDAM:
```
cd [path_to_install_directory]
```

4. Create a base folder for this install of AMSTERDAM.
```
mkdir MyAmstInstall
```

5. Move the 9 tar files from wherever they are located to your folder.
```
cd [path_to_folder_containing_tar_files]
mv *.tar [path_to_MyAmstInstall]
```

6. Extract the 9 tar files. The results are automatically put into a folder called 2009. (The second command is all one line- copy and paste it into the terminal.)
```
cd [path_to_MyAmstInstall]
tar xvfp amsterdam_4.6_v2009.part1.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part2.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part3.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part4.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part5.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part6.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part7.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part8.tar >> Extract.log ; tar xvfp
amsterdam_4.6_v2009.part9.tar >> Extract.log
```

7. Move the tar files back to where they came from:
```
mv *.tar [path_to_folder_containing_tar_files]
```

Part 2: Compiling Versions of the Libraries for Your System

Amsterdam comes with compiled versions of a number of supporting libraries it needs to run. Unfortunately, these are unlikely to be compatible with your system, unless your system happens to match that which AER used when creating Amsterdam. So, here I walk through the creation of all the necessary libraries on your system.

A) Building netCDF

AER supplied a build script and code for netCDF, but working with these things leads to problems. It is wisest to set the included version of netCDF aside and download a fresh version.

8. Navigate to the lib/ folder inside your code repository. Rename the included netCDF folder.
```
cd [path_to_MyAmstInstall]/2009/ams4.6_repository/lib
mv netCDF netCDF.IncludedWithAMST
```

9. Create a new netCDF folder. Enter it and download the compressed version of netCDF-3.6.3. Extract it.
```
mkdir netCDF
cd netCDF
wget http://www.unidata.ucar.edu/downloads/netcdf/ftp/netcdf-3.6.3.tar.gz
tar -xvf netcdf-3.6.3.tar.gz >&! extract.log
```

10. You must set a number of environmental variables to help the netCDF configure script make the right choices when setting up netCDF for your system.

If you are using a **PGF compiler**:
```
setenv FC [path_to_pgf90]
setenv F90 [path_to_pgf90]
setenv FFLAGS "-fast -Mlfs"
setenv F90FLAGS "-fast -Mlfs"
setenv CFLAGS "-O2 -DFILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE"
setenv CPPFLAGS "-DNDEBUG -DpgiFortran -DFILE_OFFSET_BITS=64 -
D_LARGEFILE_SOURCE"
```

If you are using an **iFort compiler**:
```
setenv FC [path_to_ifort]
setenv F90 [path_to_ifort]
setenv FFLAGS "-O3 -xP -ip -no-prec-div -static"
setenv F90FLAGS "-O3 -xP -ip -no-prec-div -static"
setenv CFLAGS "-O2 -DFILE_OFFSET_BITS=64 -D_LARGEFILE_SOURCE"
setenv CPPFLAGS "-DNDEBUG -DpgiFortran -DFILE_OFFSET_BITS=64 -
D_LARGEFILE_SOURCE"
```
The netCDF configure script is capable of noticing the "ifort" name in the FC / F90 variables and handling the "-DpgiFortran" flag correctly, even though we are not using pgf.

If you run into errors relating to the c portions of netCDF, you can also try setting the "CC" variable to a path to a c compiler. For me, it works best to leave the "CC" variable undefined.

11. Enter the netcdf-3.6.3 folder and run the configure script to create the makefiles which will be used to build netCDF. Be sure to set the --prefix tag equal to the base netCDF directory. (It's one level above the current working directory, but the configure script requires the whole path, not a relative path involving ".."):

```
cd netcdf-3.6.3
./configure --
prefix=[path_to_MyAmstInstall]/2009/ams4.6_repository/lib/netCDF >&!
configure.MYLOG
```

12. Run the make script to build the library.

```
make >&! make.MYLOG
```

13. Run the check functionality of the make script to make sure it was built correctly. After you do this, open up the log, go to the bottom, and make sure it says "All 7 tests passed" (around line 1164).

```
make check >&! check.MYLOG
```

14. Assuming the tests passed, run the install script to complete the installation and move the files into the right places.

```
make install >&! install.MYLOG
```

15. The install puts our netCDF library in the lib/netCDF/lib/ folder, while the Amsterdam build script expects it to be in the lib/netCDF/Linux folder. Therefore, we must change the directory name:

```
cd ..
mv lib Linux
```

B) Building ioapi

Although it is possible to build ioapi using the code included with Amsterdam, I have found it to be better to download and build a version from scratch.

16. Go to the lib/ folder in your code repository and rename the included ioapi directory. Then make a new ioapi directory.

```
cd ..
mv ioapi ioapi.IncludedWithAMST
mkdir ioapi
```

17. Enter the ioapi folder and create a directory called ioapi-3.0 inside of it.

```
cd ioapi
mkdir ioapi-3.0
```

18. Move into the ioapi-3.0 folder and download the code for ioapi. Extract it. Then enter the ioapi subdirectory.

```
cd ioapi-3.0
wget http://www.baronams.com/products/ioapi/ioapi-3.0.tar.gz
tar -xvf ioapi-3.0.tar.gz >&! extract.log
cd ioapi
```

19. We must select a Makefile and Makeinclude file. Our choice depends on the compiler we are using. Note that AMSTERDAM uses MPICH2 for handling its parallel processing, not Open MP, so we want to build the single-processor version of IOAPI whether we are making a single-processor or parallel-processor build of AMSTERDAM.

For the **PGF Compiler**:

- emacs Makefile.nocpl
  - Add **before Line 70**: BIN = Linux2_x86pg_gcc_nomp
  - Change **Line 70** to: BASEDIR = ${PWD}/..

- emacs Makeinclude.Linux2_x86pg_gcc_nomp
  - Change **Line 26** to: FC = [path_to_pgf90]
- make -f Makefile.nocpl > & ! make.nocpl.log
- cd ../Linux2_x86pg_gcc_nomp
- *You should see libioapi.a in this directory.*

For the **iFort Compiler**:

- emacs Makefile.nocpl
  - Add **before Line 70**: BIN = Linux2_x86_64ifort
  - Change **Line 70**: BASEDIR = ${PWD}/..
- emacs Makeinclude.Linux2_x86_64ifort
  - Change **Line 9** to: CC = gcc
  - Change **Line 16** to: OMPFLAGS =
  - Change **Line 17** to: OMPLIBS =
- make -f Makefile.nocpl > & ! make.nocpl.log
- cd ../Linux2_x86_64ifort
- *You should see libioapi.a in this directory.*

C) Building Stenex and/or Stenex noop

Each of these libraries consists of a main library file (with extension .a) and a number of module files (with extension .mod). The build scripts and code you need to make these are not all included with AMSTERDAM. Therefore, you'll need to download the files needed to build a vanilla version of CMAQv4.6, build the relevant libraries and modules, and move them into place:

20. Download the components of CMAQ4.6 which are necessary to build stenex, stenex_noop, and pario. In this example, we will make a directory called 4.6Download in the base of our AMSTERDAM code repository for this purpose:

```
    cd [path_to_MyAmstInstall]/2009/ams4.6_repository/
    mkdir 4.6Download
    cd 4.6Download
    wget
ftp://ftp.unc.edu/pub/cmas/pub/MODELS/CMAQ/4.6/M3MODELS.CMAQv4.6.tar.gz
    wget
ftp://ftp.unc.edu/pub/cmas/pub/MODELS/CMAQ/4.6/M3SCRIPTS.CMAQv4.6.tar.gz
```

21. Unpack them. They create the models/ (code repository) and scripts/ directories.

```
    tar -xvf M3MODELS.CMAQv4.6.tar.gz >> Extract.log ; tar -xvf
M3SCRIPTS.CMAQv4.6.tar.gz >> Extract.log
```

22. Create a "lib" directory for the final libraries to be placed in. (We are using the same structure as we would have if we were performing a normal CMAQ4.6 install, for consistency.)

```
    mkdir lib
```

23. While in the 4.6Download/ folder, set certain environmental variables (specific to each terminal window) which the script needs:

```
    setenv M3HOME $cwd ; setenv WORK $M3HOME/scripts ; setenv M3MODEL
$M3HOME/models ; setenv M3LIB $M3HOME/lib
```

24. Move to the stenex build scripts and change their permissions to make them editable:

```
    cd scripts/stenex
    chmod 775 *
```

25. There are two versions of stenex, "stenex" and "stenex_noop." Stenex_noop is used for single-processor builds, while stenex is used in parallel processor builds. However, the post-processing tools merge_concs and merge_deps require

stenex_noop, so we must always build it. To build stenex_noop:

For the **PGF Compiler**:

- emacs bldit.se_noop.pgf
- Make these edits:
  - **Line 43: Path to the pgf90 compiler**
    set FC = [path_to_pgf90]
- bldit.se_noop.pgf >&! bldit.se_noop.pgf.log

For the **iFort Compiler**:

- emacs bldit.se_noop.pgf
- Make these edits:
  - **Line 43: Path to the iFort compiler**
    set FC = [path_to_ifort]
  - **Line 44: Fortran flags**
    set FSTD = "-extend_source 132 -cm -w95 -c"
- bldit.se_noop.pgf >&! bldit.se_noop.pgf.log

26. We will now move the stenex_noop files into place in our AMSTERDAM build. A vanilla install of CMAQv4.6 keeps the stenex and/or stenex_noop output files in a single folder called "stenex". In contrast, AMSTERDAM keeps its stenex and stenex_noop files in different folders, one called "stenex" and the other called "stenex_noop". *(In case you ever have to divide up the stenex files after building both versions, stenex receives libse_snl.a, swap_sandia.mod, and the 24 .mod files which begin with "se_". Stenex_noop receives libsef90_noop.a and the 11 .mod files which begin with "noop_".)*

```
cd [path_to_MyAmstInstall]/2009/ams4.6_repository/lib/stenex_noop/
mv Linux Linux.IncludedWithAMST
mkdir Linux
cd Linux
cp
[path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/lib/stenex/Linux/* .
```

27. If you are building a **Parallel Version**, you must now create "stenex". We first rename the folder with the stenex_noop output files we made in order to prevent the two versions' output files from becoming mixed together:

For the **PGF Compiler**:

- cd [path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/lib
- mv stenex/ stenex_noop
- cd ../scripts/stenex/
- emacs bldit.se.pgf
- Make these edits:
  - **Line 43: Path to the mpich2 "include" folder (for pgf)**
    set INCL = [path_to_mpich2-103]/pgi/include
  - **Line 46: Path to the pgf90 compiler**
    set FC = [path_to_pgf90]
- bldit.se.pgf >&! bldit.se.pgf.log

For the **iFort Compiler**:

- cd [path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/lib
- mv stenex/ stenex_noop
- cd ../scripts/stenex/
- emacs bldit.se.pgf
- Make these edits:
  - **Line 43: Path to the mpich2 "include" folder (for ifort)**
    set INCL = [path_to_mpich2-103]/intel/include
  - **Line 46: Path to the iFort compiler**
    set FC = [path_to_ifort]

- **Line 47: Fortran flags**
  set FSTD = "-extend_source 132 -cm -w95 -c"
- bldit.se.pgf >&! bldit.se.pgf.log

28. If you are building a **Parallel Version**, now move the parallel stenex files into place in the AMSTERDAM build:

```
cd [path_to_MyAmstInstall]/2009/ams4.6_repository/lib/stenex/
mkdir OutOfTheWay
mv * OutOfTheWay
mkdir Linux
cd Linux
cp
[path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/lib/stenex/Linux/* .
```

D) Building Pario

29. If you are building a **Parallel Version**, move to the downloaded pario build script and change its permissions so you can edit it.

```
cd
[path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/scripts/pario
chmod 775 bldit.pario.pgf
```

30. If you are building a **Parallel Version**, edit the bldit.pario.pgf build script as follows:

For the **PGF Compiler**:

```
emacs bldit.pario.pgf
Line 41: Path to the mpich2 include folder (for pgf)
    set MPI   = [path_to_mpich2-103]/pgi/include
Line 42: Path to ioapi fixed_src directory
    set IOAPI = $M3HOME/../lib/ioapi/ioapi-3.0/ioapi/fixed_src
Line 45: Path to the pfg90 compiler
    set FC = [path_to_pgf90]
```

For the **iFort Compiler**:

```
emacs bldit.pario.pgf
Line 41: Path to the mpich2 "include" folder (for ifort)
    set MPI   = [path_to_mpich2-103]/intel/include
Line 42: Path to ioapi fixed_src directory
    set IOAPI = $M3HOME/../lib/ioapi/ioapi-3.0/ioapi/fixed_src
Line 45: Path to the iFort compiler
    set FC = [path_to_ifort]
Line 46: Fortran flags
    set FSTD = "-extend_source 132 -cm -w95 -c"
```

31. If you are building a **Parallel Version**, build pario:

```
bldit.pario.pgf >&! bldit.pario.pgf.log
```

32. If you are building a **Parallel Version**, move to the pario output directory (in the lib/ folder in your 4.6Download/ folder) and recursively change the permissions so you may copy it.

```
cd ../../lib
chmod -R 775 pario
```

33. If you are building a **Parallel Version**, in the lib/ folder of the main AMSTERDAM code repository, rename the old pario/ folder and make a copy of the pario folder from 4.6Downloads.

```
cd [path_to_MyAmstInstall]/2009/ams4.6_repository/lib
mv pario pario.IncludedWithAMST
cp -r
```

[path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/lib/pario .

E) Building m3build

The AMSTERDAM distribution comes with a build script for the m3build tool, but it lacks the necessary code to use that script. Hence, we need to make it from the version of Cmaq4.6 we downloaded.

34. Ensure that your environmental variables M3HOME, WORK, M3MODEL, and M3LIB are still correctly set relative to the 4.6Download folder (from step #23 above). Then move to the m3build build script and edit it as follows:

```
cd
[path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/scripts/build
chmod 775 bldit.m3bld
emacs bldit.m3bld
```

Line 43: Location of your C compiler
(I leave this line alone. Default is /usr/bin/gcc)

35. Build the m3build tool:

```
bldit.m3bld >&! bldit.m3bld.log
```

36. Move it into place:

```
cd ../../../lib/
mv build build.IncludedWithAMST
cp -r [path_to_MyAmstInstall]/2009/ams4.6_repository/4.6Download/lib/build
.
```

At this point, you should be using your own compiled versions for each of the items in the lib/ folder of your AMSTERDAM code repository which you are using (netCDF, ioapi, stenex_noop, build, and (if you are making a parallel version) stenex and pario).

Part 3: Creating the AMSTERDAM Application

Now we are ready to build AMSTERDAM itself. This involves building two libraries (ODEpack and Scichem), the application program (CCTM), and two post-processing tools (merge_concs and merge_deps). The main directory for building each test case is its subfolder inside [path_to_MyAmstInstall]/2009/testruns/ . In this document I use the CMAQ_AERO3_APT test case as an example, although some helpful notes about how MADRID differs are also provided.

A) Important Note on Path Length

If the path to your Amsterdam directory is too long, the creation of the application (which will be named something like CCTM_cb4_ae3_aq_AERhg_apt) will fail. The maximum path length has not been determined, but a path of 42 characters to the point where you extract the tar files was found to be too long. (That is, [42_characters]/2009 is too long.) These next two steps walk you through the creation of links which are placed at a high level of the directory structure in order to prevent any issues with excessive path length while allowing you to install Amsterdam wherever you like.

37. Go to a directory which is very high in the directory structure (a short path leading to this directory).

```
cd [short_path]/
```

38. Create links to the Amsterdam code repository and to your test case's src directory in this location. You can name the links something short and descriptive.

```
ln -s [path_to_MyAmstInstall]/2009/testruns/CMAQ_AERO3_APT/src MyCmaqAptSrc
ln -s [path_to_MyAmstInstall]/2009/ams4.6_repository MyAmstRepository
```

B) Preparing the bldit Script and Input Files

39. Gunzip all of the files in the following two directories:

```
cd [short_path]/MyCmaqAptSrc/../inp/emis/emis.bkg
gunzip *.gz &
cd ../../met
gunzip *.gz &
```

40. Go to the scripts folder (src) for the model you want to build. Move all the files except bldit into an OutOfTheWay folder:

```
cd ../../src/
mkdir OutOfTheWay
mv BLD_cb4_ae3_aq_AERhg_apt/ MOD_DIR/ bldit.log cfg.cb4_ae3_aq_AERhg_apt include/ scichem/ OutOfTheWay/
```

41. Edit bldit and make the following changes:

If you are making a version **without mercury**, change **Line 10** to:

```
set HG_FLAG = F          # T or F (simulate mercury or not)
```

If you are making a **single processor** version, comment out **Line 12:**

```
#set ParOpt        # set for multiple PE's; comment out for single PE
```

For the **PGF Compiler**:

Line 19: path to your ams4.6_repository directory

```
setenv REPOSITORY [short_path]/MyAmstRepository
```

Line 20: path to the PGF compiler

```
setenv FC [path_to_pgf90]
```

Line 26: path to to src/ directory (the current directory)

```
set Base = [short_path]/MyCmaqAptSrc
```

Line 188 (198 for MADRID): path to MPICH 2-1.0.3 folder (for pgf)

```
set MPICH = [path_to_mpich2-103]/pgi
```

(The script looks for "include/mpif.h" in the folder you specify.)

Line 207: IOAPI Directory Name for Par Configuration

```
set IOAPI_LIBDIR = ${M3LIB}/ioapi/ioapi-3.0/Linux2_x86pg_gcc_nomp
```

(This is because we always build a non-OpenMP version of IOAPI.)

Line 224 (236 for MADRID) Remove -lc flag

```
set LIBS = "$SCILIB $STENEX_LIB $LIB2 $LIB3 $MPICH_LIB $IOAPI_LIB $NETCDF_LIB  -lgcc -lgcc_eh"
```

(The -lc flag repeats the -lpthread flag added by our compiler.)

For the **iFort Compiler**:

Line 19: path to your ams4.6_repository directory

```
setenv REPOSITORY [short_path]/MyAmstRepository
```

Line 20: path to the iFort compiler

```
setenv FC [path_to_ifort]
```

Line 26: path to to src/ directory (the current directory)

```
set Base = [short_path]/MyCmaqAptSrc
```

Line 188 (198 for MADRID): path to MPICH 2-1.0.3 folder (for ifort)

```
set MPICH = [path_to_mpich2-103]/intel
```

(The script looks for "include/mpif.h" in the folder you specify.)

Line 190: Fortran Flags
set FSTD = "-extend_source 132 -cm -w95 -c"
Line 191: Fortran Flags
set F_FLAGS = "${FSTD} -O3 -module ${MODLOC} -I."
Line 207: IOAPI Directory Name for Par Configuration
set IOAPI_LIBDIR = ${M3LIB}/ioapi/ioapi-3.0/Linux2_x86_64ifort
Line 218: IOAPI Directory Name for SP Configuration
set IOAPI_LIBDIR = ${M3LIB}/ioapi/ioapi-3.0/Linux2_x86_64ifort
Line 225 (236 for MADRID) Remove -lc flag
set LIBS = "$SCILIB $STENEX_LIB $LIB2 $LIB3 $MPICH_LIB $IOAPI_LIB $NETCDF_LIB -lgcc -lgcc_eh"
(The -lc flag repeats the -lpthread flag added by our compiler.)

42. Run the build script. (This is expected to fail, but it generates files you need):
bldit >&! bldit.log01.run1

C) Making the ODEPack and Scichem Libraries

43. Navigate to the odepack makefile and change the path to the Fortran compiler:
cd scichem/odepack
emacs makefile

For the **PGF Compiler**:
Line 15:
FC = [path_to_pgf90]

For the **iFort Compiler**:
Line 15:
FC = [path_to_ifort]

44. Make the odepack library:
make >&! make.log
(If this worked, you should have the file "libodepack.a" inside your src/scichem/lib folder.)

45. Edit the make_obj_f90 file as follows:
cd ..
emacs make_obj_f90

For the **PGF Compiler**:
Line 2:
[path_to_pgf90] $1 $2

For the **iFort Compiler**:
Line 1:
echo " **** : ifort $2"
Line 2:
[path_to_ifort] $1 $2

46. If you are creating a **Version Without Mercury**, there is a Mercury-related error in the code which must be fixed at this time.
cd scipuff
emacs step_p_dyn.F90

We must edit the block of text starting at **Line 257 (270 for MADRID)**, by changing it

From:

```
if (ldark()) then
     cl2c = ccl2n(iz) * swfrac
else
     cl2c = ccl2d(iz) * swfrac
end if

hclc = chcl(iz)
```

To:

```
cl2c = 0.
hclc = 0.
```

47. If you are creating a **Version Without Mercury**, fix another error in the code which will prevent your test case from working correctly. (This fix does not affect builds containing mercury, either positively or negatively.)

```
cd [short_path]/MyCmaqAptSrc/scichem/aqueous/inc/
emacs aqueous_species_inc.F90
```

Change the block of code at **Line 28 from:**

```
#ifdef MDHGFLAG
! internal species pointers for HG0
INTEGER :: HG0
#endif
```

to:

```
! internal species pointers for HG0
INTEGER :: HG0
```

48. The variable N2O5 is initialized in two places: once in init_aqueous (aqueous species) and once in init_aero (aerosol species, which is a subset of aqueous species). This results in errors in N2O5 concentration when the "conc" and "concamb" arrays are passed from subroutine step_mc to subroutines step_aqueous and step_aerosol_chem, because there are two elements of the conc array which refer to N2O5, one as an aerosol and one as a non-aerosol species. To fix this, we need to remove N2O5 from init_aqueous, which requires changes in two files. (The first file is the same one we just edited in the previous step if making a non-Mercury build.):

```
cd [short_path]/MyCmaqAptSrc/scichem/aqueous/inc/
emacs aqueous_species_inc.F90
```

Subtract 1 from N_AQ_SPC on **lines 17 and 19**:

Line 17: INTEGER, PARAMETER :: N_AQ_SPC = 29

Line 19: INTEGER, PARAMETER :: N_AQ_SPC = 28

```
cd ..
emacs init_aqueous.F Remove N2O5 (including comma and single quotes) from
Line 43:
         &               'PACD    ','O3      ','OH      ',
```

49. There are several incorrect references to Fortran interface blocks. These must be fixed, or they may result in an Euler convergence failure when Amsterdam is run. (Even if there is no convergence failure, these bugs likely will make your results inaccurate.) There are four lines where changes are required:

```
cd [short_path]/MyCmaqAptSrc/scichem/pig/inc/
emacs interface_definitions.f90
```

Line 139:
SUBROUTINE STEP_TIME(CGRID)
Line 142:
END SUBROUTINE STEP_TIME

```
cd ..
emacs step_pig.F90
```
Line 30:
use interface_definitions, only: OUTPUT_ALL, SET_CGRID_EQM, STEP_TIME

```
emacs init_pig_inp.F90
```
Add a new line **after Line 32**:
use interface_definitions, only: OUTPUT

50. The developers have included a file called "mpif.h" in the code for AMSTERDAM. This file is actually part of AER's installation of MPICH2 which they copied into the directory [short_path]/MyCmaqAptSrc/scichem/pig/host/inc . Even though we specified the path to our own MPICH2 installation in the "bldit" script, part of the program is not written to take advantage of this. Instead, it uses AER's copy of "mpif.h," which contains settings specific to their system. Therefore, we need to replace it with a copy of the "mpif.h" file from our own MPICH2 installation.

```
cd host/inc
mv mpif.h mpif.h.IncludedWithAMST
```

If you are using the **PGF Compiler**:
```
cp [path_to_mpich2-103]/pgi/include/mpif.h .
```

If you are using the **iFort Compiler**:
```
cp [path_to_mpich2-103]/intel/include/mpif.h .
```

51. If you are building a **parallel version**, there are at least two important problems which will make your results inaccurate (or will cause your program to crash in the run phase, if you use iFort). Briefly, the problems are:

- In step_time, the program sends an array of a derived type (StepMCdat), or a sub-array of it, to the mpich2 subroutines MPI_SEND and MPI_RECV. Mpich does not support passing derived types as arguments to these functions.
- Over 50 variables which are used in subroutine step_aerosol_chem (by all processors) are initialized in subroutine init_aero, which is only executed by processor zero. (While some variables are passed to the other processors via MPI, these 50+ variables are not.)

Unfortunately, these problems are not small bugs. They are major issues relating to AER's program design. Also, given the magnitude of these errors, there are likely to be other bugs in the parallel PinG code which I did not find. My recommendation is to avoid using the parallel PinG code entirely. This can be accomplished by not executing parallel code in step.F90, which has the effect of assigning all of the puffs to be calculated by processor 0 (the main processor). This ***will significantly slow program execution***, but it is critical for accuracy and reliability of the model.

```
cd ../../../scipuff/
emacs step.F90
```

In each line which currently reads "#ifdef parallel", change the word "parallel" to a different word which is not used in the AMSTERDAM code. For example, you can change them to say "#ifdef nonsense". The lines you should change are: 35, 67, 217, 274, 303, 353, 476, and 541.

52. If you are compiling a version of Amsterdam which **uses array bounds checking** (enabled with the "-C" compiler option on PGF or the "-check bounds" compiler option on iFort), you must make the changes below in get_met.F90, where several arrays which use a second dimension of size 1 instead must be

declared with this dimension of assumed size. Note that in my testing, a run with bounds checking enabled took roughly six times as long as an identical run without bounds checking, so you should only enable array bounds checking for debugging purposes. *(This part of the code sets a passed-in one-dimensional array equal to a locally-declared two-dimensional array. This may have been intentional- for instance, an attempt to improve the program's run speed- by relying on some assumption about how these arrays are stored in memory in order to access the elements of a 1-D array via a 2-D reference system. This seems like a dangerous programming tactic, but I do not think it adversely affects the model results. The fix below doesn't alter this behavior. It merely avoids an out-of-bounds array access error by not specifying the size of the phony second dimension.)*

```
emacs get_met.F90
```

Change **Line 159** to:

```
real f(maxd,*)          !3-D variable being interpolated
```

Change **Lines 370-371** to:

```
real    aa(maxd,*)   !Part of numerator for vertical diffusivity
real    bb(maxd,*)   !Part of denominator for vertical diffusivity
```

53. Before we can compile the scichem library, we need to copy a number of variable definitions from the main bldit script to makefile.scilib.apt. This is because those definitions are remembered from bldit when bldit calls makefile.scilib.apt, but they are not remembered when we make makefile.scilib.apt ourselves. Therefore:

```
cd ..
emacs makefile.scilib.apt
```

After line 2, add the following text (but remember to modify the paths):

```
#-------------------------------------------------------------
# Variable definitions from bldit needed for scilib
#-------------------------------------------------------------
REPOSITORY = [short_path]/MyAmstRepository
```

Plus the following line if you are using the **PGF compiler:**

```
FC = [path_to_pgf90]
```

Plus the following line if you are using the **iFort compiler:**

```
FC = [path_to_ifort]
```

Plus the following line if you are building a **parallel version:**

```
PARALLEL_FLAG = -Dparallel
```

Plus the following line if you are building a **version with mercury:**

```
APT_HG_FLAG = -DMDHGFLAG
```

Plus one of the following lines depending on which mechanism you are using:

CMAQ-APT, with mercury

```
Mechanism = cb4_ae3_aq_AERhg
```

CMAQ-APT, no mercury

```
Mechanism = cb4_ae3_aq_AERXhg
```

MADRID, with mercury

```
Mechanism = cb4_aq_hg_aeMADRID_2sec
```

MADRID, no mercury

```
Mechanism = cb4_aq_Xhg_aeMADRID_2sec
```

54. If you are building an **iFort version**, you must also make the following changes to the Fortran flags in makefile.scilib.apt:

Around Line 75: Length flag

```
LENGTHFLAG = -extend_source 132
```

On each of the following 8 lines (spread across roughly Lines 78 - 95): FLAGS0, FLAGS1, FLAGS2, FLAGSA, FLAGSP, FLAGS3, FLAGSH, FLAGSH2

- Remove the "-Mstandard" and "-fast" flags from each line. (FLAGS0, FLAGSH, and FLAGSH2 have no "-fast" flag to remove.)
- Add "-cm -w95 -O3" to each line.

55. Compile the schichem library:

```
make -f makefile.scilib.apt >&! make.scilib.log
```

(If this worked, you should have the file "libsci.a" inside your src/scichem/lib folder.)

D) Building AMSTERDAM CCTM

Unfortunately, there are a number of bugs which affect certain builds. These issues will cause errors when AMST is run, but they must be fixed now in the build phase.

56. Go to the src/ folder. We need to make two changes to the bldit script. We want to set it to produce a makefile, and we should comment out the lines which will cause it to replace the files we fixed (for odepack and scilib) with not-fixed versions from the repository. Commenting out these lines also avoids the need to change the permissions on scichem/ (and on subsequent runs of the build script, include/).

```
cd ..
emacs bldit
```

Uncomment Line 61 (In MADRID, it's Line 63 and it's already done.)

```
set MakeOpt            # builds a Makefile to make the model
```

Comment Out Lines 87, 89, and 95 (In MADRID, 89, 91, & 97) Those are:

```
# cp -p -r $M3MODEL/CCTM/src/ping/ping_apt_AE3/scichem .
# make >& make.log
# make -f makefile.scilib.apt >& make.scilib.log
```

57. Run the build script (to generate a makefile):

```
bldit >&! bldit.log01.run2
```

58. Go into the BLD_cb4_ae3_aq_AERhg_apt/ directory. (Your directory name will be slightly different if you used MADRID or a no-Mercury build.)

```
cd BLD_cb4_ae3_aq_AERhg_apt/
```

59. If you are building a **parallel version**, change the permissions so distr_env.c is writeable, then edit it as follows (this file doesn't exist if you're making a single-processor version):

```
chmod 775 distr_env.c
emacs distr_env.c
```

Lines 45 and 46:

```
#define TEMP_BUF_SIZE   40960
#define CURR_STR_SIZE   4096
```

60. AER left the file AERO_EMIS.F in a state which is suitable for neither a mercury nor a no-mercury build. It contains code which causes an error in a no-mercury build, but for a mercury build, the code to read particulate mercury (PHGI) from the background emissions is commented out. *(This was likely done because PHGI does not exist in the included background emissions files, so commenting it out avoids a run-time error. However, this section of code is needed to properly set variable VPHG, which is used to set EPHG, a variable used in sections of the code which are not commented out. By default, Fortran is likely assigning the value zero to the uninitialized variable EPHG, thereby*

generating results which omit some or all background aerosol mercury emissions.)

This problem can be fixed by noting that there is a variable in the background emissions files, AHGPJ, which should only differ from PHGI by about 0.1% in terms of mass emission rate (based on the parameters FACEM25_ACC and FACEM25_ATKN, which are used to divide up particulate mercury by mode). Therefore, AHGPJ is a suitable replacement. In mercury builds, we change the code to use AHGPJ when PHGI is not available. *(Note that AHGPJ is already read from the AE_EMIS table elsewhere in this function for a different purpose, so it will be used twice following this fix. This is not a problem.)* You need to make the file writable and make one of these changes:

```
chmod 775 AERO_EMIS.F
emacs AERO_EMIS.F
```

If you are building a **version without mercury**, use the character "C" to comment out **lines 712 - 728** in **AERO_EMIS.F**. Be sure to put the "C"s in the first column.

```
C            VNAME = 'AHGPJ'
C            N = INDEX1( VNAME, N_AE_EMIS, AE_EMIS )
C            IF ( N .NE. 0 ) THEN
C               VHGPJ = N
C            ELSE
C               XMSG = 'Could not find ' // VNAME // 'in AE_EMIS table'
C               CALL M3EXIT ( PNAME, JDATE, JTIME, XMSG, XSTAT3 )
C            END IF
C
C            VNAME = 'AHGPI'
C            N = INDEX1( VNAME, N_AE_EMIS, AE_EMIS )
C            IF ( N .NE. 0 ) THEN
C               VHGPI = N
C            ELSE
C               XMSG = 'Could not find ' // VNAME // 'in AE_EMIS table'
C               CALL M3EXIT ( PNAME, JDATE, JTIME, XMSG, XSTAT3 )
C            END IF
```

If you are building a **version with mercury**, do not make the change above. Instead, replace the commented-out code from **lines 869 - 882** with the following code block:

```
C mercury
C Edited so that if PHGI is not found, AHGPJ is used in its place
            VNAME ='PHGI'
            INDX = INDEX1( VNAME, NVARS3D, VNAME3D )
            IF ( INDX .NE. 0 ) THEN
               V = V + 1
               VPHG = V
               AEMIS( V ) = VNAME
               UNITSAE( V ) = UNITS3D( INDX )
            ELSE
               VNAME ='AHGPJ'
               INDX = INDEX1( VNAME, NVARS3D, VNAME3D )
               IF ( INDX .NE. 0 ) THEN
                  V = V + 1
                  VPHG = V
                  AEMIS( V ) = VNAME
                  UNITSAE( V ) = UNITS3D( INDX )
               ELSE
                  XMSG = 'Could find neither PHGI nor '
     &            // VNAME( 1:TRIMLEN( VNAME ) )
     &            // ' in aerosol table'
                  CALL M3EXIT ( PNAME, JDATE, JTIME, XMSG, XSTAT3 )
```

```
                  END IF
               END IF
```

61. If you are building a **version without mercury**, use the character "C" to comment out **lines 1002 - 1011** in **aero_driver.F**. Be sure to put the "C"s in the first column. You first need to make the file writeable.

```
chmod 775 aero_driver.F
emacs aero_driver.F
```

```
C         VNAME = 'HG2'
C         N = INDEX1( VNAME, N_GC_G2AE, GC_G2AE )
C         IF ( N .NE. 0 ) THEN
C            LHG2  = GC_STRT - 1 + GC_G2AE_MAP( N )
C            MWHG2 = GC_MOLWT( GC_G2AE_MAP( N ) )
C         ELSE
C            XMSG =
C     &         'Could not find ' // VNAME // 'in gas chem aerosol table'
C            CALL M3EXIT ( PNAME, JDATE, JTIME, XMSG, XSTAT3 )
C         END IF
```

62. If you are building a **parallel version**, there is a bug in ping_apt.F which can cause a problem with communication between processors, resulting in nonsensical output data. (This fix has no effect on a single-processor version.) The bug can be fixed via:

```
chmod 775 ping_apt.F
emacs ping_apt.F
```

Line 69: GCSIZE variable declaration

```
      INTEGER, SAVE :: GCSIZE      ! Message size of global array
```

63. **NOTE:** In **MADRID**, the CCTM application gets created by bldit, despite the fact that MakeOpt was set. This executable does not include the code changes we just made. Therefore, after making the fixes described above, you need to:

If building a **parallel version of MADRID**:

```
rm distr_env.o
rm ping_apt.o
```

If building a **version of MADRID without mercury**:

```
rm aero_driver.o
```

Always when building MADRID:

```
rm AERO_EMIS.o
rm ../MOD_DIR/aero_emis.mod
rm CCTM_cb4_aq_hg_aeMADRID_2sec_apt
```

64. Use the makefile to create the Amsterdam application:

```
make >&! make.log01
```

If this step worked, you should see an application named "CCTM_cb4_ae3_aq_AERhg_apt" (or similar, if you used a different chemical mechanism) in your current folder.

Part 4: Building the merge_concs and merge_deps Post-Processing Tools

---

If the last step in Part 3 (above) is successful, the main CMAQ application is built. However, the build script likely failed to compile two supporting tools: merge_concs and merge_deps. Use the following procedure to build them manually.

65. Enter the scichem/ directory and edit makefile.merge_concs.

```
cd ../scichem
```

emacs makefile.merge_concs

After Line 2, paste the following text:

```
#-------------------------------------------------------------
# Variable definitions from bldit needed for merge_concs
#-------------------------------------------------------------
REPOSITORY = [short_path]/MyAmstRepository
M3LIB = $(REPOSITORY)/lib
```

Plus the following line if you are using the **PGF compiler:**

```
FC = [path_to_pgf90]
```

Plus the following line if you are using the **iFort compiler:**

```
FC = [path_to_ifort]
```

Plus one of the following lines depending on which mechanism you are using:

CMAQ-APT, with mercury

```
Mechanism = cb4_ae3_aq_AERhg
```

CMAQ-APT, no mercury

```
Mechanism = cb4_ae3_aq_AERXhg
```

MADRID, with mercury

```
Mechanism = cb4_aq_hg_aeMADRID_2sec
```

MADRID, no mercury

```
Mechanism = cb4_aq_Xhg_aeMADRID_2sec
```

66. Also, make the following changes to makefile.merge_concs. *(Note that the post-processing tools cannot be statically linked on Emerald, so I remove the "-Bstatic" flag in order to dynamically link them. If you wish to see if you can create a statically-linked version of merge_concs on your system, try leaving the -Bstatic flag in place and perhaps remove the -lc flag from line 60. This can also be attempted in a 32-bit environment, which may have the libraries necessary to statically link merge_concs even if they are lacking in your 64-bit environment.)*

If using the **PGF Compiler**:

Line 54: Change the path to the MPICH2 lib folder

```
MPICH = [path_to_mpich2-103]/pgi/lib
```

If using the **iFort Compiler**:

Line 54: Change the path to the MPICH2 lib folder

```
MPICH = [path_to_mpich2-103]/intel/lib
```

Always:

Line 76: Remove the -Bstatic flag

```
LINK_FLAGS = -Wl
```

67. Create merge_concs by running the makefile:

```
make -f makefile.merge_concs >&! make.merge_concs.log
```

If this step worked, you should see "merge_concs" in your scichem/bin/ folder.

68. Complete all of the other steps in Part 4 for "merge_deps" in order to create that tool.

Part 5: Running the Included Test Case in AMSTERDAM

The AMSTERDAM .tar files come with a set of input files (emissions, boundary

conditions, initial conditions, etc.). You should try running the included test case on your system so that you can make sure your install of AMSTERDAM is working, and also to give you standardized results which can be compared against those obtained by AER and those we have at the UNC Institute for the Environment.

This run procedure assumes you use LSF (Load-Sharing Facility) to handle your jobs. LSF uses the "bsub" command. If you do not use LSF/bsub, then you will need to modify these directions to suit your own research cluster.

69. Navigate to the sim/ directory inside your base test case folder. Move everything except for the run script, the GRIDDESC.ALGA12 file, and the in_out.q file out of the way:

```
cd [path_to_MyAmstInstall]/2009/testruns/CMAQ_AERO3_APT/sim
mkdir OutOfTheWay
mv LOGS/ machines4 master.cmaq.apt.job mpd.hosts readme.mpd readme.run
OutOfTheWay
```

70. Edit the run.cmaq.job.apt script and make the following changes:

```
emacs run.cmaq.apt.job
```

If you wish to simulate **Day 183**:

Line 3: Julian Date

```
set jdate = 2002183
```

After Line 3, add the line:

```
setenv aptrestart F
```

If you wish to simulate **Day 184**:

Line 3: Julian Date

```
set jdate = 2002184
```

After Line 3, add the line:

```
setenv aptrestart T
```

(When aptrestart is false, CCTM believes this is the first day of a simulation and therefore it needs just a cmaq.RESTART.[date] file in the prior day's output directory. When aptrestart is true, CCTM believes this is a subsequent day of a multi-day run, and it expects a complete set of output files in the prior day's output directory.)

If you are performing a simulation **Without Mercury**, change **Line 6** to:

```
 set APPL     = cb4_ae3_aq_AERXhg_apt
```

For a Parallel Build

Replace everything after Line 237 ("ls -l $SRCDIR/$EXEC; size $SRCDIR/$EXEC") with:

```
        #> Instructions to use multiple (4) processors and produce a
logfile
        #BSUB -n 4
        #BSUB -oo run.AMST.log.job=%J
        #BSUB -R "xeon30 span[ptile=4] same[bc_chassis]"
        #BSUB -J AMST_run
        #BSUB -a mpich2
        #BSUB -q week

        mpirun.lsf $BASE/$EXEC

        exit()
```

For a Single-Processor Build

Line 9: Processor Domain Breakdown

```
setenv NPCOL_NPROW "1 1"
```

Line 10: Number of Processors

```
set NPROCS   = 1
```

Replace everything <u>after</u> Line 237 ("ls -l $SRCDIR/$EXEC; size $SRCDIR/$EXEC") with:

```
#> Instructions to use a single processor and produce a logfile
#BSUB -n 1
#BSUB -oo run.AMST.log.job=%J
#BSUB -R xeon30
#BSUB -J AMST_run
#BSUB -a mpich2
#BSUB -q week

mpirun.lsf $BASE/$EXEC

exit()
```

71. Make an alias for the CCTM application inside your sim folder. We no longer need to use the links we created at [short_path] (see section 3A, "Important Note on Path Length"), but you may do so if you are confident that these links will not be deleted. (Note that the names of the BLD folder and CCTM application may be slightly different in your build.)

```
ln -s
[path_to_MyAmstInstall]/2009/testruns/CMAQ_AERO3_APT/src/BLD_cb4_ae3_aq_AERhg_a
pt/CCTM_cb4_ae3_aq_AERhg_apt
```

72. If you are simulating **<u>Day 183</u>**, move aside the included output files for day 183.

```
cd ../out/2002183
mkdir IncludedWithAMST
mv *.* IncludedWithAMST
```

73. If you are simulating **<u>Day 184</u>**, gunzip the compressed output files for day 183 supplied by AER and move aside the included output files for day 184.

```
cd ../out/2002183
gunzip *
cd ../2002184
mkdir IncludedWithAMST
mv *.* IncludedWithAMST
```

74. Move back to the sim/ folder and submit the run script as an LSF job:

```
cd ../../sim
bsub < run.cmaq.apt.job
```

NOTE: Before each run, you need to delete or move aside any output files the program puts into the out/2002183/ or out/2002184 folders (depending on which day you are simulating).

Part 6: Running the merge_concs and merge_deps Post-Processing Tools

---

The merge_concs tool operates on a CONC.200218x output file. At each timestep, it merges all puffs into the concentration grid, CGRID. The resulting file, CONC_merged.200218x, will show higher concentrations than CONC.200218x because visualization programs such as PAVE and VERDI cannot see the puffs.

75. First, we move AER's logs and output files out of the way so they are not over-written.

```
cd [path_to_MyAmstInstall]/2009/testruns/CMAQ_AERO3_APT/pp/merge_concs
```

```
    mv out out.IncludedWithAMST
    mkdir out
    cd sim
    mv LOGS LOGS.IncludedWithAMST
    mkdir LOGS
```

76. The script to run merge_concs is merge_concs.job. If you wish to run merge_concs on a single day using LSF (bsub), you should create a separate file to run merge_concs.job (rather than editing merge_concs.job and adding #BSUB directives). This is because the line which calls the executable in merge_concs.job uses redirect symbols to generate a log file which BSUB incorrectly interprets as applying to the BSUB command.
    emacs run.merge_concs (creates a new file)
    Paste the following text into the file:

```
#! /bin/csh -f
setenv JULDATE 2002183

#BSUB -n 1
#BSUB -oo merge_concs.log.job=%J
#BSUB -R xeon30
#BSUB -J mrg_cncs
#BSUB -q week

./merge_concs.job
```

The script merge_concs.job itself does not appear to need any edits, as long as all files are in the expected places (as is true if this guide is followed precisely).

77. Submit the new run script to bsub.
    bsub < run.mergeconcs

A log file for the run is placed in the LOGS/ folder, while a merged CONC output file is placed in the ../out/ folder.

78. Complete all of the other steps in Part 6 for "merge_deps" in order to run that tool.

ERROR FAQ

---

netCDF Errors

Q1: The **configure script for netCDF (using iFort)** fails to find C-equivalent to Fortran routine "SUB"
A1: You used the -fast flag in one of the FF or F90 lines. This flag only works for PGF builds. On iFort, use the -O3 flag instead.

ioapi Errors

Q2: When compiling **ioapi**, I get the error:

```
/for_main.o: In function `main':
/export/users/nbtester/efi2linux_nightly/branch-
11_0/20081106_010000/libdev/frtl/src/libfor/for_main.c:(.text+0x38): undefined
reference to `MAIN__'
/tmp/ifort40RMmz.o: In function `currec_':
currec.F:(.text+0x78): undefined reference to `nextime_'
currec.F:(.text+0x9a): undefined reference to `nextime_'
currec.F:(.text+0x103): undefined reference to `nextime_'
[... a number of similar lines ...]
```

currec.F:(.text+0x25a): undefined reference to `nextime_'
make: *** [currec.f] Error 1
A2: This error appears to happen intermittently and can be "fixed" without making any changes. If you get this error, try removing all files from the output directory,
[path_to_MyAmstInstall]/2009/ams4.6_repository/lib/ioapi/ioapi-3.0/Linux2_[ending varies based on build type]. Then make ioapi again.

scilib Errors

Q3: The **scilib makefile** generates the error "PGF90-S-0038-Symbol, hg0, has not been explicitly declared (mpitype.f90)"
A3: There are two possible reasons for this bug:
Cause 1) You are running a simulation with mercury, but you did not correctly add "APT_HG_FLAG = -DMDHGFLAG" into makefile.scilib.apt before first running it.
Cause 2) If you are making a parallel, no-mercury build, you have forgotten to fix the bug in aqueous_species_inc.F90 described above. You need to remove the #ifdef MDHGFLAG and #endif lines from the block of code starting at Line 28.

Q4: When I try to compile **scilib** using the makefile, I get an error like this:
PGF90-S-0017-Unable to open include file: GC_SPC.EXT (host_chem_inc.F: 9)
PGF90-S-0017-Unable to open include file: AE_SPC.EXT (host_chem_inc.F: 10)
... and 5 more lines which are similar, followed by 3 lines complaining about a non-constant expression.
A4: You did one (or both) of the following things wrong:
Cause 1) The makefile did not understand the "Mechanism" you used. You must include a line setting the mechanism variable, and it must match the string the build script would have set precisely. This string varies depending on whether or not you are including mercury and whether you are building CMAQ-APT or MADRID.
Cause 2) You accidentally used a different version of the PGF compiler to build scilib after using pgf v9.0.1 to run the build script for the first time (or some other PGF version mismatch). Use the same compiler throughout.

Q5: When I try to compile **scilib** using the makefile, I get an error like this:
PGF90-F-0004-Unable to open MODULE file common_met.mod (amb_data_pig.f90: 20)
(This error may name any module file, not necessarily the one in the example above.)
A5: You have deleted (or failed to create) a module file, while the associated .o (compiled output) file of the same name still exists. In order for the makefile to recreate a deleted or renamed module file, the associated .o file must also be renamed or deleted. Locate and delete the relevant .o file manually, or use the included clean.job script to delete all modules and all output files. (Note that on some systems, including Emerald, you must modify the clean.job script such that the first line is split into four lines, each with its own \rm command.)

Q6: When making **scilib (using iFort)**, I get a bunch of errors like:
error #5082: Syntax error, found END-OF-STATEMENT when expecting one of: * , /
DATA GC_SPC(30), GC_MOLWT(30) / 'CRES ', 108.0 /
----------^
A6: You forgot to change line 75 of the makefile.scilib.apt to use the -extend_source 132 option.

CCTM errors (build-time)

Q7: The **CCTM build script or CCTM makefile** reports errors in VGRD_DEFN.F on lines 91, 92, and 93.
A7: The path to your install folder is too long. Use links as described above

(near the beginning of the section on "Creating the AMSTERDAM Application") to shorten it.

Q8: From the **CCTM build script or CCTM makefile**, I am getting errors about multiple definitions of '__lll_lock_wait_private'
and '__lll_unlock_wait_private'.
A8: You need to remove the -lc flag from the "set LIBS" line in your build script, as it is being duplicated by a flag added by the compiler.

Q9: From the **CCTM build script or makefile**, I am getting an error in ch3u_connect_sock.c which says, "warning: Using 'gethostbyname' in statically linked applications requires at runtime the shared libraries from the glibc version used for linking"
A9: This is a warning, not an error. It will not prevent the CCTM application from compiling or running. Ignore it.

Q10: When building **CCTM**, there was an error when opening a module file. For example, the error may look like this:

```
PAGRD_DEFN.F(91): error #7002: Error in opening the compiled module
file.  Check INCLUDE paths.   [NOOP_MODULES]
         USE NOOP_MODULES       ! stenex
-------------^
compilation aborted for PAGRD_DEFN.F (code 1)
```

A10: One or more module files needed by the makefile are missing from MOD_DIR. Note that the stenex, stenex_noop, and pario bldit scripts can all finish building their respective libraries (.a extension) without having correctly built all the modules. If this happens, the missing modules are not copied into MOD_DIR. One thing which can cause some of the .MOD files to fail to build is if you are using the iFort compiler, and you have forgotten to update the Fortran flags lines ("set FSTD = ...") in the stenex, stenex_noop, or pario build scripts.

Q11: When building **CCTM (using iFort)**, I get undefined reference errors in the function 'ping_' like this:

```
/tmp/ipo_ifort6SAsCe.o: In function `ping_':
/tmp/ipo_ifort6SAsCe.f:(.text+0x59f07): undefined reference to `step_pig_'
/tmp/ipo_ifort6SAsCe.f:(.text+0x5aae6): undefined reference to `init_scichem_'
```

A11: You are using the -fast (or -ipo) compiler option for optimization. Use the -O3 compiler option instead.

Q12: When building **CCTM (using iFort)**, I get undefined reference errors in various functions inside IOAPI which refer to "--kmpc_global_thread_num" , "__kmpc_critical" , and "__kmpc_end_critical".
A12: The "-openmp" flag (for multithreading) was used when building IOAPI. AMSTERDAM does not rely on IOAPI Open MP functionality, even for parallel processor builds. Remove the -openmp flags from the Makeinclude file when building IOAPI.

Run-Time Errors

Q13: When I run CCTM, I get an error which says "your temp_buf in distr_env may not big enough to hold next environmental pair".
A13: When you built the Amsterdam CCTM application (parallel version), you did not sufficiently increase the size of the buffers in distr_env.c. You need to go back, set Amsterdam to use a makefile, increase the buffer size, and make the file.

Q14: When I run CCTM, it outputs data only for the first hour (00:00:00). The run log includes several (4) copies of the following error:

```
    ******ERROR******
     Routine=open_new_mcfile
      Error opening SCICHEM file
```

File=[path_to_MyAmstInstall]/2009/testruns/CMAQ_AERO3_APT/sim/../out/2002183/cmaqapt.dgn
Make sure file does not already exist

A14: This error may indicate that you have forgotten to remove or rename the output files in the out/2002183/ directory before running CCTM. **However**, if you are certain that you have moved or renamed these files, then this error instead indicates that you are using a parallel version of CCTM with a single-processor version of scilib. This can happen if you did not correctly define the environmental variable "PARALLEL_FLAG = -Dparallel" in the scilib makefile.

Q15: When I run CCTM, I get the following error:

******ERROR******
Routine=init_dmp_file
Error opening SCICHEM-PiG 3D PiG transferred-to-host file
File=[path_to_MyAmstInstall]/2009/testruns/CMAQ_AERO3_APT/sim/../out/2002183/cmaqapt.dmp

A15: Make no changes and run CCTM again. This error has only ever occurred for me on the first run of CCTM and resolves itself thereafter.

Q16: When I run Amsterdam, I get an error which says, "Cannot find /opt/mpich2//bin/mpiexec. Exiting ...".
A16: This is a problem which was caused by LSF (bsub). It occurs when the cluster uses a fresh (unmodified) version of the LSF software, so the problem is re-introduced whenever ITS upgrades to a new LSF version. ITS needs to fix the LSF scripts, the mpich2 wrapper, or both. (They have never informed me about exactly what they do to solve this problem.)

Q17: When I run CCTM, I get the following error:

*** ERROR ABORT in subroutine RDEMIS_AE on PE 000
Could not find AHGPJ in AE_EMIS table
Date and time 0:06:00 July 2, 2002 (2002183:000600)

A17: This error occurs if you are using a no-mercury build, but you failed to comment out lines 712 - 728 in AERO_EMIS.F. You need to delete the CCTM application, AERO_EMIS.o, and AERO_EMIS.mod. Then make the necessary change to the code and re-build the application.

Q18: When I run CCTM, I get the following error:

*** ERROR ABORT in subroutine AERO_DRIVER
Could not find HG2 in gas chem aerosol table
Date and time 0:00:00 July 2, 2002 (2002183:000000)

A18: This error occurs if you are using a no-mercury build, but you failed to comment out lines 1002 - 1011 in aero_driver.F. You need to delete the CCTM application and aero_driver.o. Then make the necessary change to the code and re-build the application.

Q19: When I run a parallel build of CCTM, my run terminates abruptly after outputting lists of emissions from all 40 plume-in-grid sources. (Each source is separated in the log by a header which says "get_emissions:".) iFort may call this error a **segmentation fault**, while PGF may simply say there was a **collective abort of all ranks**.
A19: There is a problem in an MPICH call in the subroutine step_time in the file src/scichem/scipuff/step.F90. One likely source of this problem is that your build used AER's version of the src/scichem/pig/host/inc/mpif.h file. You should replace this with a copy of mpif.h built for your own system. (Note that the parameter MPI_ADDRESS_KIND should equal 8 on a 64-bit system and 4 on a 32-bit system. You may wish to raise the buffer value of MPI_BSEND_OVERHEAD from AER's choice of 59 to 95.)

Q20: When I run a parallel iFort build of CCTM, the run freezes (possibly at timestep 3:12 or timestep 4:24). No error is printed to the main output log, but one of the other three processors' logs (CTM_LOG_001, CTM_LOG_002, or CTM_LOG_003) has the error:

*** ERROR ABORT in subroutine AQCHEM on PE 001

Maximum AQCHEM total iterations exceeded

A20: This problem may occur because scilib is attempting to send an array of a derived type to MPI_SEND and MPI_RECV. Use the workaround presented above to ensure that all puffs are handled by the main processor.

Q21: When I run a parallel build of CCTM, my run completes successfully, but the output netCDF CONC.2002183 data file does not make sense. It appears that concentrations change normally in the lower left quadrant of the map, while the other three quadrants only show minor changes, appearing mostly "frozen" in many timesteps.

A21: This is a problem with processor communication. You may have forgotten to add the "SAVE" attribute to the GCSIZE variable in file ping_apt.F. (Note that if you use iFort, this fix may cause the run to crash if you have not also fixed the problem in scilib where an array of a derived type is used as an argument for MPI_SEND and MPI_RECV.)

Q22: When I run CCTM in a with-mercury test case which was built with **array bounds checking enabled** (the "-C" option in PGF, the "-check bounds" option in iFort), I get the following error from each processor:

```
0: Subscript out of range for array pm_em (AERO_EMIS.F: 1162)
    subscript=0, lower bound=1, upper bound=6, dimension=1
```

A22: This error occurs if VPHG is not set because the code for reading PHGI was commented out by AER, and VPHG remains set to zero or an unpredictable number. You must complete the fix for AERO_EMIS.F described in part 3D above.

Q23: When I run CCTM in a test case which was built with **array bounds checking enabled** (the "-C" option in PGF, the "-check bounds" option in iFort), I get the following error (or a similar error mentioning array "aa" or array "bb"):

```
0: Subscript out of range for array f (get_met.F90: 181)
    subscript=5, lower bound=1, upper bound=1, dimension=2
```

A23: You have not changed the second dimension of arrays f, aa, and bb in src/scichem/scipuff/get_met.F90 to use an assumed size. Complete the relevant step in part 3C above to resolve these errors.

Appendices

A) Guide to Dependencies

In the course of troubleshooting errors, you may make a change to a component and wonder which other components need to be rebuilt because they rely on that component.

If you change and re-build:	Then you must also rebuild:
netCDF	CCTM, merge_concs, merge_deps
IOAPI*	CCTM, merge_concs, merge_deps
Stenex	CCTM
Stenex_noop	CCTM, merge_concs, merge_deps
Pario	CCTM
odepack	CCTM
scilib	CCTM
CCTM	nothing
merge_concs, merge_deps	nothing

* Note that scilib relies on the IOAPI fixed_src directory but not on any compiled ioapi files. Therefore, unless you are changing the code of IOAPI in the fixed_src directory, you do not have to rebuild scilib when you rebuild IOAPI.

If you change this:	Then you must rebuild (or build):
compiler type	everything
mpich2 version	stenex, pario, CCTM

Mercury vs. No Mercury	scilib, CCTM, merge_concs, merge_deps
SP to Parallel	stenex, pario, scilib, CCTM
Parallel to SP**	scilib, CCTM
test case (CMAQ-APT vs. MADRID)	odepack, scilib, CCTM, merge_concs, merge_deps

** In this case, you must also build stenex_noop if you have not already done so for the post-processing tools.

B) Version History of This Document

Version 1.03 - Fixed issue with whitespace in step 62. Changed step 63 to apply to all MADRID builds. Fixed typos.

Version 1.02 - Added step 48 to fix a problem with N2O5. Renumbered later steps accordingly. Edited step 51 to use a more efficient work-around to improve runtime. Improved the explanation of aerosol mercury problem in step #60.

Version 1.01 - Added Part 6 on running merge_concs and merge_deps. Expanded step 50 to indicate there are multiple serious problems with the parallel PinG code, and the work-around is now the recommended permanent solution (barring the development of an improved version of AMSTERDAM). Added step 51 to address three errors in versions build with array bounds checking enabled (and renumbered later steps). Updated step 59 to include a critical fix for aerosol mercury in mercury builds. Added Q22 and Q23 to cover errors in builds with array bounds checking enabled.

Version 1.00 - Added steps 50 and 60 to work around problems in parallel builds. Added related FAQ entries Q20 and Q21. Added folder rename to step 27. Fixed paths in steps 29, 49. Small improvements to steps 19, 36, 47, 49, 61, and FAQ A2.

Version 0.96 beta - Added numbering to Error FAQ. Fixed two small errors (in step 25 and FAQ A1).

Version 0.95 beta - First release provided to others. Parallel builds still have an unresolved issue. Procedures here should be sufficient for single-processor builds.

References for Chapter 1

Air New Zealand, 2010. Aircraft Statistics. http://www.airnewzealand.co.nz/aircraft-statistics. Accessed 4-10-11.

Airports Council International, 2010. *Passenger Traffic 2009 Final*. http://www.airports.org/cda/aci_common/display/main/aci_content07_c.jsp?zn=aci&cp=1-5-54-55-9812_666_2. Accessed 2-16-11.

Arunachalam, S., B.H. Baek, A. Holland, Z. Adelman, F.S. Binkowski, A. Hanna, T. Thrasher and P. Soucacos, 2008. *An Improved Method to Represent Aviation Emissions in Air Quality Modeling Systems and their Impacts on Air Quality,* In Proceedings of the 13th Conference on Aviation, Range and Aerospace Meteorology, New Orleans, LA, Jan 2008.

Arunachalam, S., B Wang, N Davis, BH Baek, and JI Levy, 2011. Effect of Chemistry-Transport Model Scale and Resolution on 1 Population Exposure to $PM_{2.5}$ from Aircraft Emissions during Landing and Takeoff**,** *Atmospheric Environment*, doi:10.1016/j.atmosenv.2011.03.029.

Baek, B.H., S. Arunachalam, A. Holland, Z. Adelman, A. Hanna, T. Thrasher, and P. Soucacos (2007) *Development of an Interface for the Emissions Dispersion and Modeling System (EDMS) with the SMOKE Modeling System*. In Proceedings of the 16th Annual Emissions Inventory Conference - "Emissions Inventories: Integration, Analyses and Communication*",* Raleigh, NC, May 2007.

Byun, D.W. and J.K.S. Ching, ed., Science Algorithms of the EPA Models-3 Community Multiscale Air Quality (CMAQ) Modeling System, 1999, EPA/600/R-99/030(peer reviewed). [Available from National Exposure Research Laboratory, RTP, NC]

Byun, D.W., and Schere, K.L., 2006. Review of the Governing Equations, Computational Algorithms, and Other Components of the Models-3 Community Multiscale Air Quality (CMAQ) Modeling System. *J. Applied Mechanics Reviews*, 59 (51), 2006.

Cimorelli, A., S. Perry, A. Venkatram, J. Weil, R. Paine, R. Wilson, R. Lee, W. Peters, R. Brode, 2005. AERMOD: A dispersion Model for Industrial Source Applications. Part I: General Model Formulation and Boundary Layer Characterization. *Journal of Applied Meteorology* 44, I 5 p 682-693.

Community Multiscale Air Quality (CMAQ) Modeling System, 2011. http://www.cmaq-model.org

CSSI, Inc., 2007. Project 16: Investigation of Aviation Emissions Air Quality Impacts Technical Report. Unpublished.

Eder, B., Yu, S., 2005. A performance evaluation of the 2004 release of Models-3 CMAQ. *Atmospheric Environment* 40, 4811-4824.

Environmental Protection Agency. Alternative Models. http://www.epa.gov/scram001/dispersion_alt.htm#scipuff. Accessed 10/31/10.

Eyers, C., Gilboy, M., 2007. Revision of Calvert method for filling in missing smoke number data. QinetiQ. Unpublished.

Federal Aviation Administration, 2010. Aviation Environmental Design Tool (AEDT). U.S. Department of Transportation. March 10, 2010. http://www.faa.gov/about/office_org/headquarters_offices/aep/models/aedt/. Accessed 10-25-10.

FAA and EPA, 2009. Recommended Best Practice for Quantifying Speciated Organic Gas Emissions from Aircraft Equipped with Turbofan, Turbojet, and Turboprop Engines. http://www.faa.gov/regulations_policies/policy_guidance/envir_policy/media/FAA-EPA_RBP_Speciated%20OG_Aircraft_052709.pdf. Accessed 2-28-11.

Federal Register, 1998. Emissions and Dispersion Modeling System (EDMS) policy for airport air quality analysis: Interim guidance to FAA Orders 1050.1D and 5050.4A, 63 Federal Register 70 (30 April 1998), p. 18068.

Gladstone, C., Glover, G., Massimini, P., Skiotsuki, C., Simmons, B., 2000. Analysis of Triple Arrivals to Hartsfield Atlanta International Airport. MITRE Corporation. http://www.caasd.org/library/tech_docs/2000/mtr00w0000023.pdf. Accessed 3-26-11.

Hall, C., S. Mondoloni, T. Thrasher, 2003. Estimating the Impact of Reduced Thrust Takeoff on Annual NOx Emissions at Airports. *CSSI, Inc.* http://www.cssiinc.com/public/technicalpapers/docs/13-Estimating%20the%20Impact%20of%20Reduced%20Thrust%20Takeoff%20on%20Annual%20NOx%20Emissions.pdf. Accessed 2-16-11.

Herndon, C., Jayne, J., Lobo, P., Onasch, T., Fleming, G., Hagen, D., Whitefield, P., Miake-Lye, R., 2008. Commercial Aircraft Engine Emissions Characterization of in-Use Aircraft at Hartsfield-Jackson Atlanta International Airport. *Environmental Science & Technology 42*, 1877-1883.

Industrial Economics, 2006. Expanded Expert Judgment Assessment of the Concentration-Response Relationship between $PM_{2.5}$ Exposure and Mortality. Cambridge, MA: Prepared for the Office of Air Quality Planning and Standards, U.S. Environmental Protection Agency. http://www.epa.gov/ttn/ecas/regdata/Uncertainty/pm_ee_report.pdf. Accessed 4-7-11.

International Civil Aviation Organization (ICAO), 1993. International Standards and Recommended Practices – Environmental Protection. Volume II. Second Edition.

International Civil Aviation Organization (ICAO), 2010. Emissions Databank. Issue 17. http://www.caa.co.uk/default.aspx?catid=702. Accessed 2-28-11.

Karamchandani, P., Seigneur, C., Vijayaraghavan, K., Wu, S.Y., 2002. Development and application of a state–of–the–science plume–in–grid model. *Journal of Geophysical Research D: Atmospheres* 107, art. No. 4403.

Karamchandani, P., Vijayaraghavan, K., Chen, S., Seigneur, C., Edgerton, E., 2006. Plume-in-grid modeling for particulate matter. *Atmospheric Environment* 40, 7280-7297.

Karamchandani, P., Vijayaraghavan, K., Chen, S., Balmori-Bronson, R., Knipping, E., 2010. Development and application of a parallelized version of the advanced modeling system for transport, emissions, reactions and deposition of atmospheric matter (AMSTERDAM): 1. Model performance evaluation and impacts of plume–in–grid treatment. *Atmospheric Pollution Research 1* 260-270.

Levy, J., Hsu, H., Melly, S., 2008. High Priority Compounds Associated with Aircraft Emissions: PARTNER 11 final report on subtask: Health Risk Prioritization of Aircraft Emissions Related Air Pollutants. *Partnership for AiR Transportation Noise and Emissions Reduction*. http://web.mit.edu/aeroastro/partner/reports/proj11/p11compndsemiss.pdf . Accessed 4-10-11.

Levy, JI, M. Woody, BH Baek, U. Shankar and S. Arunachalam (2011) Current and Future Particulate Matter-related Mortality Risks from Aviation Emissions in the United States, *Risk Analyses*, Under Review.

Moss, M., H. Segal, 1994. Emissions and Dispersion Modeling System: Its Development and Application at Airports and Airbases. *Journal of the Air and Waste Management Association* 44, I 6.

Noel, G., R. Cointin, D. Allaire, S. Jacobson, K. Willcox, 2009. Assessment of the Aviation Environmental Design Tool. *8th USA/Europe Air Traffic Management Research and Development Seminar.* http://www.atmseminarus.org/seminarContent/seminar8/papers/p_084_EI.pdf. Accessed 10/25/10.

Pope C, Burnett R, Thun M, Calle E, Krewski D, Ito K, Thurston G, 2002. Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution. JAMA; 287(9):1132-1141.

Ratliff, G., Sequeira, C., Waitz, I., Ohsfeldt, M., Thrasher, T., Graham, M., Thompson, T., 2009. Aircraft Impacts on Local and Regional Air Quality in the United States. PARTNER Project 15 final report. http://web.mit.edu/aeroastro/partner/reports/proj15/proj15finalreport.pdf . Accessed 3-24-11.

Rice, C., 2003. Validation of Approach and Climbout Times-in-Mode for Aircraft Emissions Computation. *2003 Transportation Research Board Annual Meeting.* www.ltrc.lsu.edu/TRB_82/TRB2003-001446.pdf. Accessed 2-28-11.

Sage Management, 2010. SCIPUFF Dispersion Model. http://www.sage-mgt.net/services/modeling-and-simulation/scipuff-dispersion-model . Accessed 3-24-11.

Santos, L. Sykes, R., Karamchandani, P., Seigneur, C., Lurmann, F., Arndt, R., Kumar, N., 2000. Second-order Closure Integrated Puff (SCIPUFF) model with gas and aqueous phase chemistry and aerosols. 11th Joint Conference on the Applications of Air Pollution Meteorology with the Air and Waste Management Association.

Spicer, C. W., M. W. Holdren, R. M. Riggin and T. F. Lyon, (1994). Chemical composition and photochemical reactivity of exhaust from aircraft turbine engines. *Annales Geophysicae* 12:944-955.

Timko, M., Onasch, T., Northway, M., Jayne, J., Canagaratna, M., Herndon, S., Wood, E., Miake-Lye, R., 2010. Gas Turbine Engine Emissions–Part II: Chemical Properties of Particulate Matter. *Journal of Engineering for Gas Turbines and Power* 132.

Tsia, F., Sun, W., Chen, J., 2001. A Composite Modeling Study of Civil Aircraft Impacts on Ozone and Sulfate over the Taiwan Area. *Journal of Terrestrial, Atmospheric, and Oceanic Sciences.* 12(1).

Vijayaraghavan, K., Karamchandani, P., Seigneur, C., Balmori, R., Chen, S., 2008. Plume-in-grid modeling of atmospheric mercury. *Journal of Geophysical Research* 113.

Wayson, R.L., Fleming, G.G., Kim, B., 2003. Status report on proposed methodology to characterize jet/gas turbine engine particulate matter emissions. U.S. Department of Transportation, Federal Aviation Administration.

Wayson, R., Fleming, G., Iovinelli, R., 2009. Methodology to Estimate Particulate Matter Emissions from Certified Commercial Aircraft Engines. *Journal of the Air & Waste Management Association* 59, 91-100.

Wilkerson, J., M. Jacobson, A. Malwitz, S. Balasubramanian, R. Wayson, G. Fleming, A. Naiman, S. Lele, 2010. Analysis of Emission Data from Global Commercial Aviation: 2004 and 2006. *Amospheric Chemistry and Physics.* http://www.atmos-chem-phys.net/10/6391/2010/acp-10-6391-2010.pdf. Accessed 10-25-10.

Wong, H., P. Yelvington, M. Timko, T. Onasch, R. Miake-Lye, 2008. Microphysical Modeling of Ground-Level Aircraft-Emitted Aerosol Formation: Roles of Sulfur-Containing Species. *Journal of Propulsion and Power* 24(3).

Wood, E., Herndon, S., Timko, M., Yelvington, P., Miake-Lye, R., 2008. Speciation and chemical evolution of nitrogen oxides in aircraft exhaust near airports. *Environ Sci Technol* 43(6), 1884-91.

Woody, M. and S. Arunachalam, 2010. Secondary Organic Aerosol Produced from Aircraft Emissions at 1 the Atlanta Airport: An Advanced Diagnostic Investigation Using Process Analysis, In Preparation.

Woody, M., B.H. Baek, Z. Adelman, M. Omary, Y.-F. Lam, J. West and S. Arunachalam, 2011. An Assessment of Aviation Contribution to Current and Future Fine Particulate Matter in the United States, *Atmospheric Environment,* Accepted.

UK Department for Transport, 2006. Project for the Sustainable Development of Heathrow: Air Quality Technical Report. http://www.dft.gov.uk/stellent/groups/dft_aviation/documents/divisionhomepage/612123.hcsp

Unal A., Hu Y., Chang M.E., Odman M.T., Russell A.G., 2005. Airport related emissions and impacts on air quality: Application to the Atlanta International Airport. *Atmospheric Environment* 39(32), 5787-5798.

US EPA, 2010. 2002 National Emissions Inventory Data and Documentation. http://www.epa.gov/ttn/chief/net/2002inventory.html . Accessed November 2010.

II. Equity and Health Impacts of Aircraft Emissions at the Hartsfield-Jackson Atlanta International Airport

Chapter Abstract

This study examined the impacts of aircraft emissions on $PM_{2.5}$ concentrations during the months of June and July, 2002 at the Hartsfield-Jackson Atlanta International Airport, the world's busiest airport in terms of passenger traffic. Primary and secondary pollutants were modeled using the Advanced Modeling System for Transport, Emissions, Reactions, and Deposition of Atmospheric Matter (AMSTERDAM). Geographic Information System (GIS) analysis was used to apportion modeled concentrations to census tracts within the 29-county Atlanta region. A statistical analysis was conducted to determine population exposure and to investigate correlations between pollutant concentrations and demographic variables (median household income, race, educational attainment, and home value).

It was found that in 675 census tracts, the average aircraft contribution to $PM_{2.5}$ was below 20 ng/m^3. These census tracts reflected all levels of income, home values, racial compositions, and levels of educational attainment. However, aircraft contribution to $PM_{2.5}$ exceeded 20 ng/m^3 in 37 census tracts. These tracts overwhelmingly tended to have lower median income and home values, as well as higher percentages of nonwhite residents and residents with low educational attainment. These relationships are statistically significant at the 99.8% level.

Analysis based on a concentration-response function indicates that if June and July 2002 are representative of the entire year in terms of aircraft impacts, the Atlanta airport's influence on $PM_{2.5}$ concentrations is responsible for 1.4 premature adult deaths in that year.

Introduction and Background

The Hartsfield-Jackson Atlanta International Airport is the busiest airport in the world in terms of passenger traffic, with 88 million passengers in 2009 (Airports Council International, 2010). Hartsfield-Jackson Airport is a major contributor to the economy of the Atlanta region, generating over 434,000 jobs and $58.2 billion in business revenue, counting direct, indirect, and induced impacts (Hartsfield-Jackson, 2009). However, major airports like Hartsfield-Jackson can have substantial adverse impacts on the surrounding communities in the forms of noise and air pollution (Waitz, 2004).

The economic benefits attributable to airports have been well studied (Button and Lall, 1999). A major airport provides direct employment to thousands of people and spurs demand for nearby services (such as hotels and warehousing), which provide additional jobs. An industry-accepted rule of thumb states that there are likely to be approximately 1000 direct jobs for each 1 million passengers or cargo work load units (WLU, equivalent to 100 kg of cargo) per year (Graham, 2008).

For example, a recent study of the Taiwan Taoyuan International Airport, which moves just 22 million passengers per year, found that the airport indirectly generates 1.06 jobs for each directly generated job, resulting in a total contribution of 46,823 jobs and a total economic benefit of €913 million for the region in 2008 (Lu, 2011). In addition,

major airports catalyze business and tourism activity and provide tax revenue for state and local governments (Graham, 2008).

The effect of noise from airports on nearby residents has also been well established (Swift, 2010) and has led to a number of laws and programs designed to minimize adverse impacts. The Aviation Safety and Noise Abatement Act of 1979 (49 U.S.C. § 47501 et seq.) established a methodology to determine human exposure to airport noise and specified certain planning documents, including noise exposure maps, be submitted to the FAA by participating airports (Papsidero, 1992). Many large airports, including Atlanta Hartsfield, participate in an FAA program which funds the noise insulation of properties within the contours of the official noise exposure map for the airport, as shown in Figure 14 (Hartsfield-Jackson, 2011).

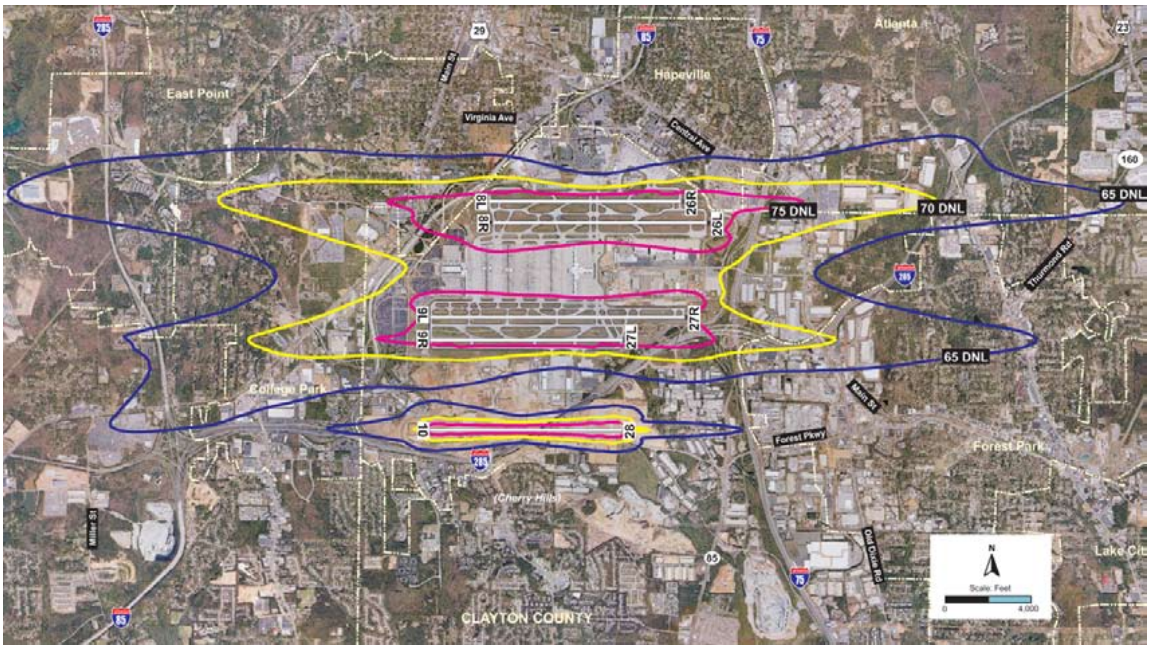

Figure 14 - Hartsfield-Jackson noise exposure map (Hartsfield-Jackson, 2011)

The FAA emphasizes the need for land use compatibility planning for the areas surrounding airports, identifying four "key issues" which must be addressed: aircraft

noise, nearby tall structures, electronic interference with navigation equipment, and interactions between aircraft and wildlife (FAA, 2011).

However, the impacts of aircraft emissions on communities near to major airports, as well as on the larger metropolitan region, are often not taken into account in the land use planning process. This is unfortunate, as atmospheric modeling studies have shown that airports can be significant contributors to local air pollution (Ratliff et al., 2009; Arunachalam et al., 2008, 2011). Exposure to air pollutants, such as particulate matter 2.5 microns or less in diameter ($PM_{2.5}$), has been associated with increased rates of lung cancer and cardiopulmonary mortality (Pope et al., 2002). It is important to understand the level of pollutant exposure caused by airports (and airport expansion projects), particularly busy airports located in densely populated regions, so that planners can better understand the possible public health implications and consider interventions to mitigate air quality impacts.

In this study, we used a recently developed atmospheric model to characterize the impact of aircraft emissions near Hartsfield-Jackson Atlanta International Airport on pollutant concentrations throughout the 29-county Atlanta region. The program, the Advanced Modeling System for Transport, Emissions, Reactions, and Deposition of Atmospheric Matter (AMSTERDAM) (Karamchandani et al., 2010), tracks emitted pollutants, changes pollutant concentrations due to chemical reactions occurring in the atmosphere, transports pollutants via winds, and accounts for physical processes such as coagulation and deposition of particulate matter. AMSTERDAM tracks emissions from aircraft as distinct "puffs," or three-dimensional volumes inside which pollutants are confined. These puffs move inside a three-dimensional grid that represents the

background air, including the emissions from all non-aircraft sources. Chapter I describes the technical aspects of the model in depth and compares the methodology used in this study with previous modeling efforts.

We used the results of our modeling analysis of changes in $PM_{2.5}$ concentration caused by aircraft activity to perform a population exposure analysis, obtaining information about the contribution of aircraft to the pollutant concentrations in communities across the 29-county Atlanta region. We also performed a statistical analysis to determine if aircraft emissions differentially impact people based on their income, race, or educational attainment. These results may have implications for environmental justice and equity. Finally, we estimate potential health impacts as a result of our calculated exposure levels based on a concentration-response function that relates all-cause, adult mortality to average $PM_{2.5}$ concentrations. We consider several policy and planning strategies that may reduce exposure to harmful air pollutants at new and existing airports.

We believe our results have relevance beyond the Atlanta airport. The Atlanta airport was studied because of its large size and data availability, but our methods could be applied to model emissions from other airports (as well as the projected increase in emissions due to future airports and airport expansion projects) to better understand their health and equity impacts. Emissions from other airports are likely to cause health and equity impacts in a similar manner to those observed in this study. Air quality modeling may be a valuable tool with which planners and local governments can predict the likely impacts from newly proposed airports or airport expansion projects. Similarly, a

demographic analysis could inform planners and local officials of the health costs and social equity impacts associated with existing airports or planned airport projects.

Methodology

Assessing air quality impacts due to the Atlanta airport involved three main tasks, including adaptation of the AMSTERDAM model for use in modeling emissions from Hartsfield-Jackson Airport, creation of emissions input files which represent aircraft emissions, and subsequent statistical and spatial analysis of model output.

All aircraft emissions within the lowest 3000 feet of altitude (part of the landing and take off cycle, or LTO cycle) for the months of June and July, 2002 were included. This date range was selected because this is the period for which model input data was available. Aircraft emissions were categorized by engine power setting: **taxi** (or **idle**) emissions which occur on the ground, **take off** emissions which occur from zero to 1000 feet for departing planes, **climb out** emissions which occur from 1000 to 3000 feet for departing planes, and **approach** emissions from 3000 to zero feet for arriving planes (Rice, 2003). (Data on emissions during the **cruise** mode was not available.) Differences due to the various models of engine used by aircraft as well as the variable level of aircraft activity during the modeling period were accounted for.

Two simulations were used to determine the effect of aircraft on pollutant concentrations. In the base test case, all emissions were included, and in the second test case, aircraft emissions were omitted. The model produced results specifying the concentration of various pollutants in each hour of the simulation. In each 4 km grid cell,

ground level concentrations from the simulation without aircraft were subtracted from concentrations in the simulation with aircraft to determine aircraft-specific impacts.

Demographic data for the census tracts in the Atlanta region (*N=712*) was obtained from the most recent release of the Atlanta Region Information System (ARIS) Volume 1c, a publicly available data package produced by the Atlanta Regional Commission for use in planning and research activities (Atlanta Regional Commission, 2011). This dataset includes the complete 28-county Atlanta metropolitan statistical area (MSA) plus Hall County.

GIS analysis (ArcGIS 9.3) was used to determine the average aircraft contribution to pollutant concentrations in each census tract. To alleviate the problem of multiple grid cells overlapping individual census tracts, a weighted average of cell pollutant values was applied to census tract polygons (zonal statistics operation). Figure 15 shows the average aircraft contribution to $PM_{2.5}$ in the 29-county Atlanta region in June-July 2002 by census tract.

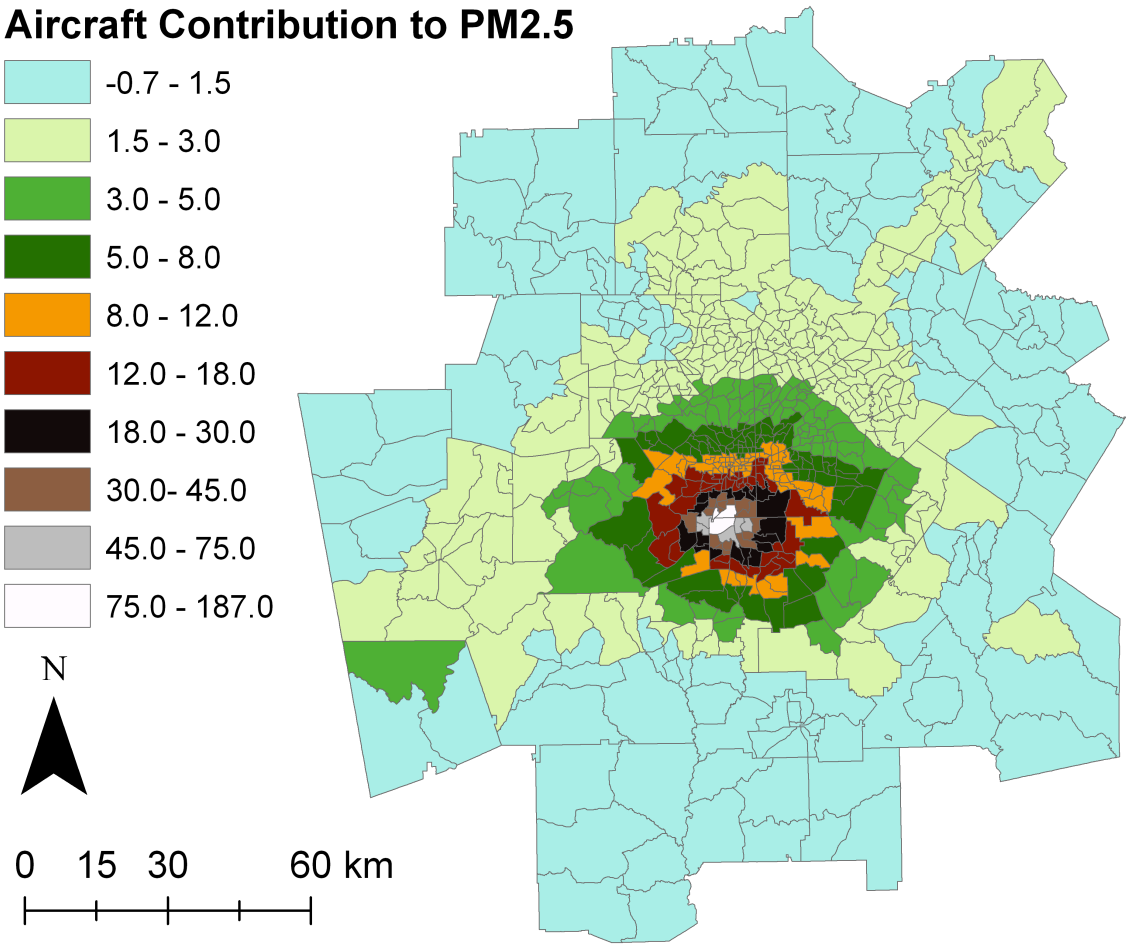

Figure 15 - Aircraft contribution to average $PM_{2.5}$ concentrations (ng/m^3) in census tracts in the 29-county Atlanta region in June and July, 2002

Finally, a statistical analysis was performed to determine the relationship between a census tract's pollutant contribution from aircraft and four demographic variables from the 2000 census: median household income, percentage of the population which is nonwhite, percentage of the adult population without a high school diploma, and median home value (U.S. Census, 2000).

Results and Analysis

In the analysis below, it is important to keep in mind that these figures show aircraft contribution to total $PM_{2.5}$ levels, time-averaged across all of June and July, 2002.

These values are low compared to the EPA's national ambient air quality standards for $PM_{2.5}$ (15 $\mu g/m^3$ annual average, 35 $\mu g/m^3$ daily average) (U.S. EPA) because those standards refer to total $PM_{2.5}$ concentration, and aircraft are only responsible for a small part of total $PM_{2.5}$.

Although average aircraft contribution has a roughly circular pattern around the airport (Figure 15), aircraft contributions in any particular hour typically form a plume, which extends away from the airport in a particular direction (Figure 16). This direction changes with the wind, so a tract's average concentration is a function of both its proximity to the airport as well as how often the wind was blowing toward that census tract during the modeling period. Census tracts beneath a plume suffer aircraft contribution to $PM_{2.5}$ in that hour which is substantially higher than the average concentration values assigned to census tracts (Figure 15) and used in the statistical analysis below.

Additionally, individuals are most active and outdoors in the daytime, which is when most flights occur. For all of these reasons, particular individuals are likely to be exposed to higher concentrations of aircraft-derived $PM_{2.5}$ than average modeled concentrations might suggest.

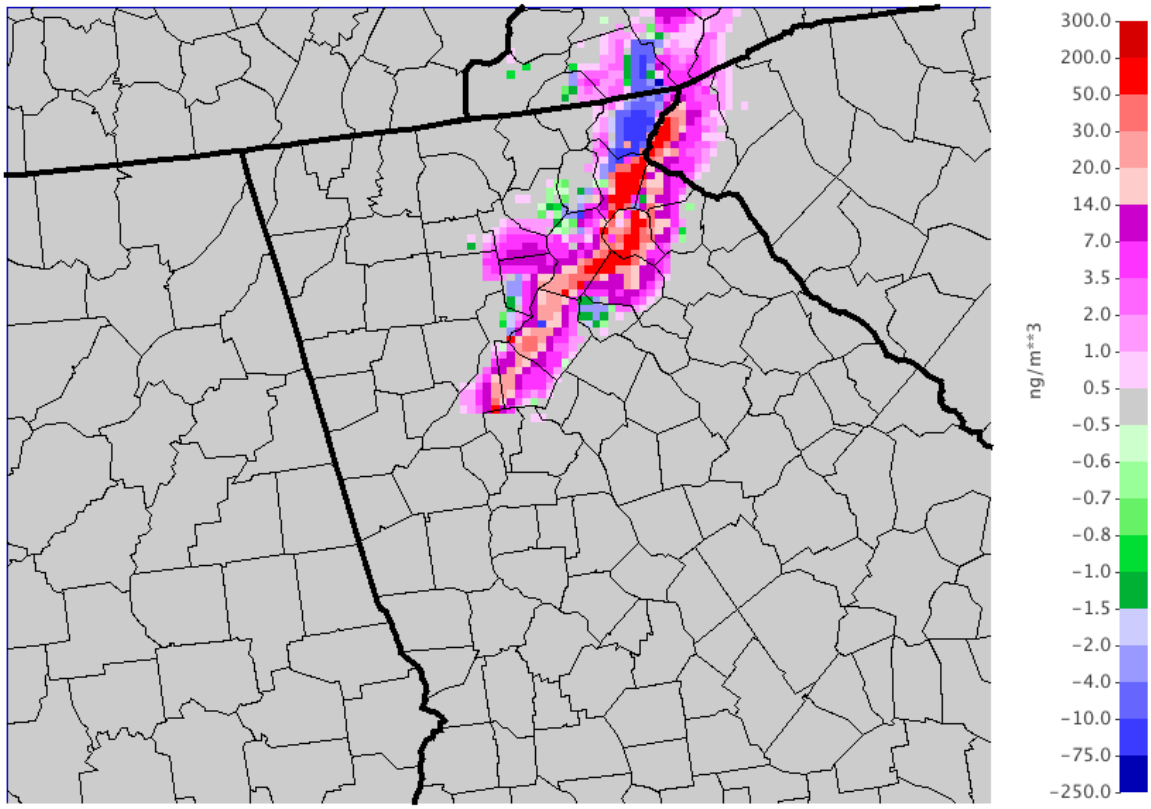

Figure 16 - Aircraft contribution to ground-level $PM_{2.5}$ concentrations (ng/m^3) at 6:00 pm on June 6, 2002

Figure 17 is a histogram depicting population exposed to various levels of aircraft-contributed total $PM_{2.5}$. Over 2.8 million people were exposed to additional average levels of particulate matter between 2 and 10 ng/m^3. Over 200,000 individuals were exposed to an increase of more than 20 $\mu g/m^3$. Excluding one outlier where only 18 people lived (the tract containing the airport itself), the maximum observed increase due to aircraft was 187 ng/m^3.

In a few census tracts, slight decreases due to aircraft were observed. This is because in some outlying areas, aircraft reduce sulfate aerosol concentrations. The physical processes behind this reduction are discussed in Chapter I.

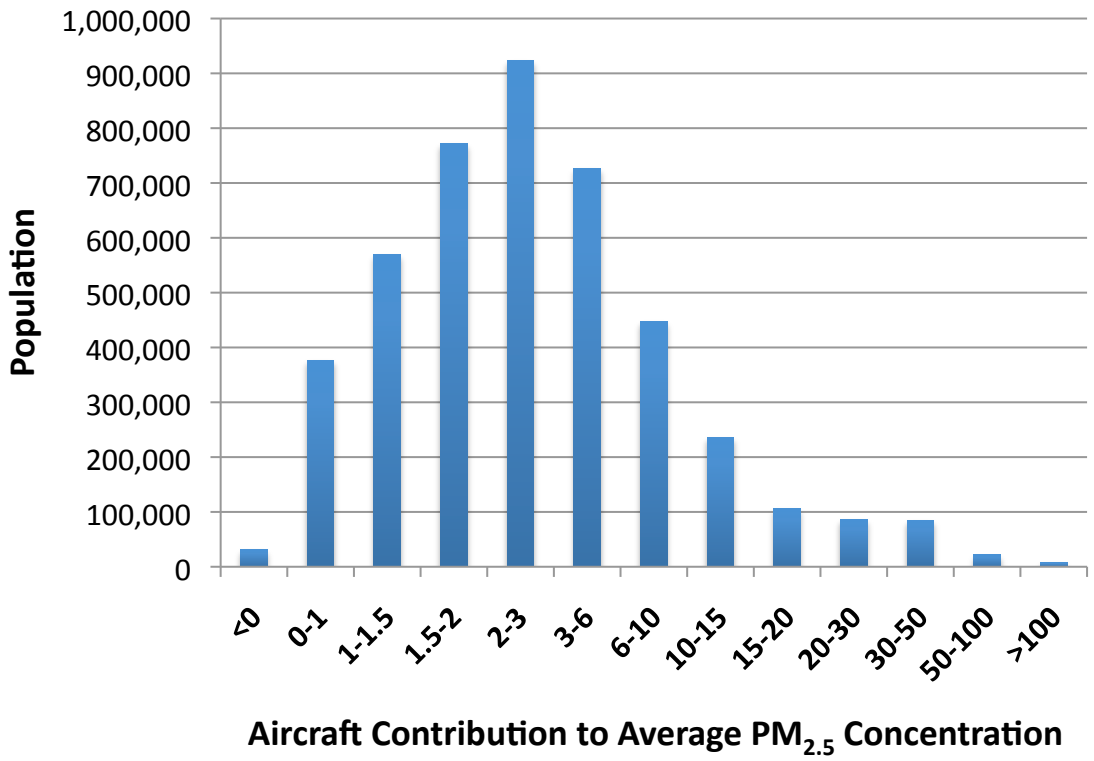

Figure 17 - Population exposure to $PM_{2.5}$ (ng/m^3) attributed to aircraft, averaged over June and July, 2002

In our analysis pertaining to education, race, and income, two outlier census tracts were excluded, including the tract containing the airport (because it possessed only 18 residents, an order of magnitude fewer residents than any other tract) and a tract reporting zero average household income and zero average home value. The analysis pertaining to median home value excluded these tracts as well as five additional tracts reporting zero average home value.

Figure 18 shows the relationship between median household income for each census tract and aircraft contribution to $PM_{2.5}$ in that tract. At low levels of aircraft contribution, there exist census tracts with a wide range of median incomes. However, every tract in which aircraft contributed at least 27 ng/m^3 to pollutant levels had a median income below the 50^{th} percentile. Every census tract in which aircraft contributed at least

32 ng/m^3 was in the lowest quartile (25%). Table 5 shows the tracts in each income quartile divided into three exposure categories: under 20 ng/m^3, 20-50 ng/m^3, and over 50 ng/m^3. The significance of the relationship between these variables is tested via Fisher's exact test (Table 5). (The chi-square test was not used because some categories in the analysis had insufficiently many values – i.e., less than 5.) We find the relationship between aircraft contribution to $PM_{2.5}$ and median household income is significant at the 99.9% level.

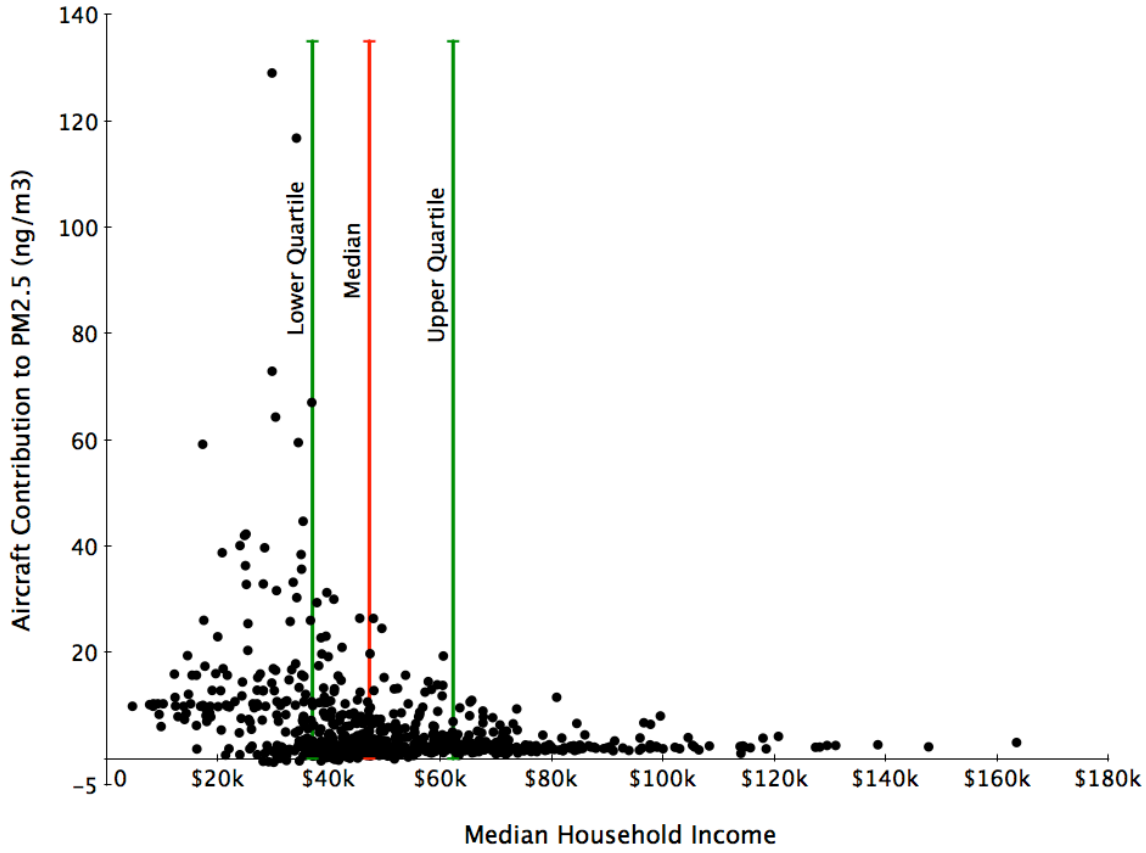

Figure 18 - Median household income vs. aircraft contribution to average $PM_{2.5}$ in June and July 2002 for individual census tracts in the 29-county Atlanta metropolitan area

	Contrib From Aircraft			
HH Income	Under20ng	20to50ng	Over50ng	Total
LowestQuartile	151	20	7	178
SecondQuartile	170	7	0	177
ThirdQuartile	175	2	0	177
UpperQuartile	178	0	0	178
Total	674	29	7	710

Fisher's exact = 0.000

Table 5 - Median household income vs. aircraft contribution to average $PM_{2.5}$ in June and July 2002 evaluated with Fisher's exact test for statistical significance

Figure 19 shows the relationship between the percentage of the population in each census tract that is nonwhite and aircraft contribution to total $PM_{2.5}$ concentrations. For this purpose, we use the Atlanta Region Information System (Atlanta Regional Commission, 2011) definition of nonwhite, which is simply the total population of the census tract minus those who selected only "White" on their census forms.

We observe that there are census tracts with a wide range of racial compositions that experience little $PM_{2.5}$ contribution from aircraft. However, all tracts with an aircraft contribution greater than 12 ng/m^3 had a higher percentage of nonwhite residents than the median census tract. More than half of these tracts were in the upper quartile. Table 6 depicts tracts divided into the same three exposure categories used for median income in Table 5, yielding a relationship between race and exposure that is significant at the 99.9% level.

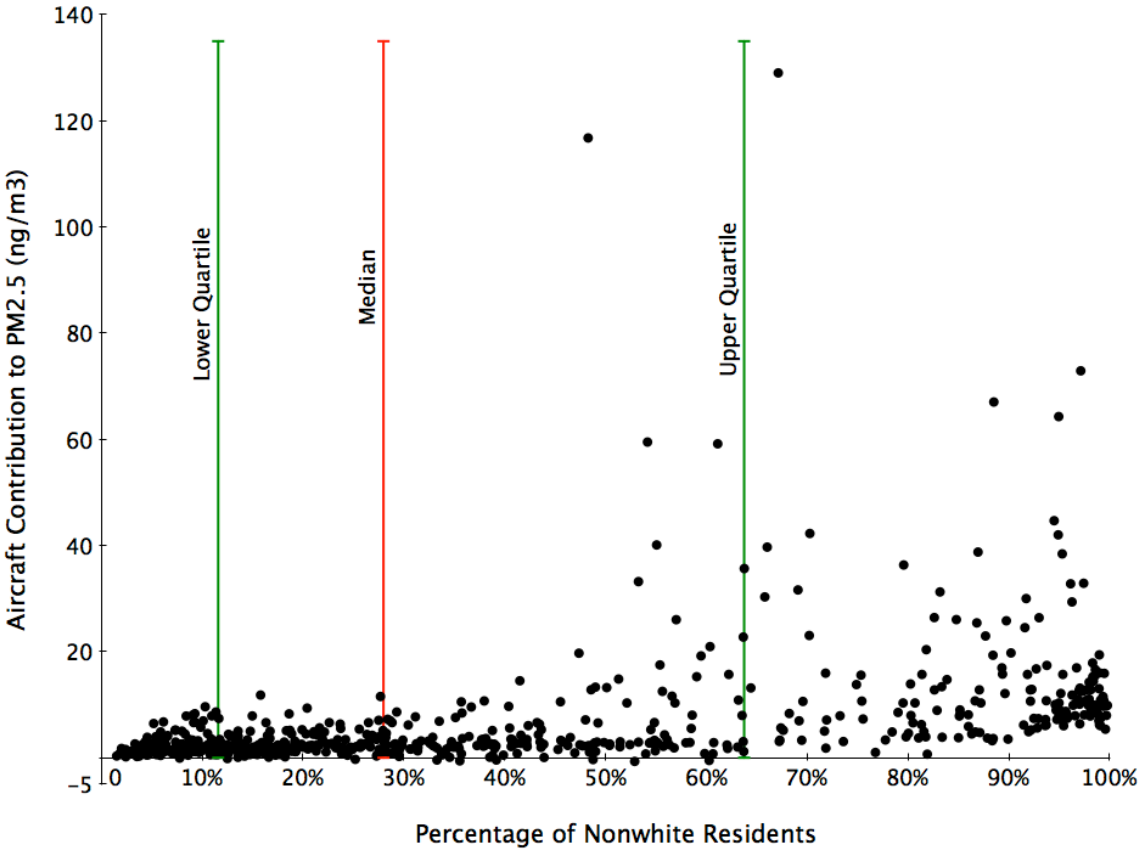

Figure 19 - Percentage of population that is nonwhite vs. aircraft contribution to average $PM_{2.5}$ in June and July 2002

	Contrib From Aircraft			
% Nonwhite	Under20ng	20to50ng	Over50ng	Total
LowestQuartile	178	0	0	178
SecondQuartile	177	0	0	177
ThirdQuartile	169	5	3	177
UpperQuartile	150	24	4	178
Total	674	29	7	710

Fisher's exact = 0.000

Table 6 - Percentage of population that is nonwhite vs. aircraft contribution to average PM2.5 in June and July 2002 evaluated with Fisher's exact test for statistical significance

Figure 20 shows the percentage of residents of each tract age 25 or higher without a high school diploma (or equivalent) vs. aircraft contribution to total $PM_{2.5}$. As before, census tracts with low aircraft contribution fell across the spectrum in terms of education. However, census tracts with aircraft contribution above 20 $\mu g/m^3$ overwhelmingly fell above the median, and many high pollutant values were in the upper quartile. Table 7 shows the tracts grouped by education quartile and exposure category, with the statistical significance evaluated via Fisher's exact test. The relationship is significant at the 99.8% confidence level. The percentage of residents without a high school diploma is a slightly weaker predictor of high aircraft $PM_{2.5}$ contributions than the other metrics examined.

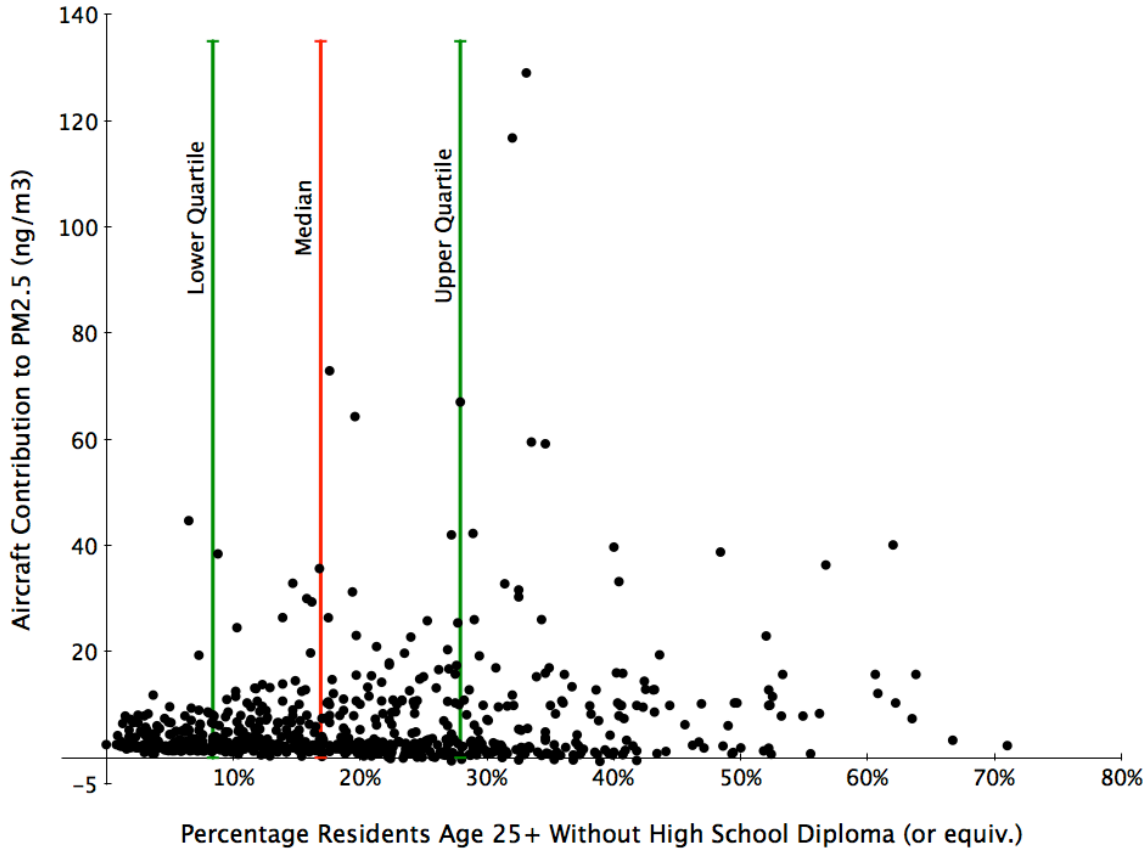

Figure 20 - Percentage of population age 25+ without a high school diploma (or equiv.) vs. aircraft contribution to average $PM_{2.5}$ in June and July 2002

	Contrib From Aircraft			
% No HS Diploma	Under20ng	20to50ng	Over50ng	Total
LowestQuartile	171	1	0	172
SecondQuartile	174	7	0	181
ThirdQuartile	167	9	2	178
UpperQuartile	162	12	5	179
Total	674	29	7	710

Fisher's exact = 0.002

Table 7 - Percentage of population age 25+ without a high school diploma (or equiv.) vs. aircraft contribution to average PM2.5 in June and July 2002 evaluated with Fisher's exact test for statistical significance

Figure 21 shows the relationship between median home value and aircraft contribution to total $PM_{2.5}$. As was the case with the other demographic variables, we see tracts with the complete range of home values that experienced very little aircraft contribution to $PM_{2.5}$. However, all but one census tract all census tracts with an aircraft contribution of greater than 14 ng/m^3 possessed an average home value below the median, and most of these census tracts had average home values in the lowest quartile. In Table 8 we see that the relationship between exposure and median home value is significant at the 99.9% confidence level.

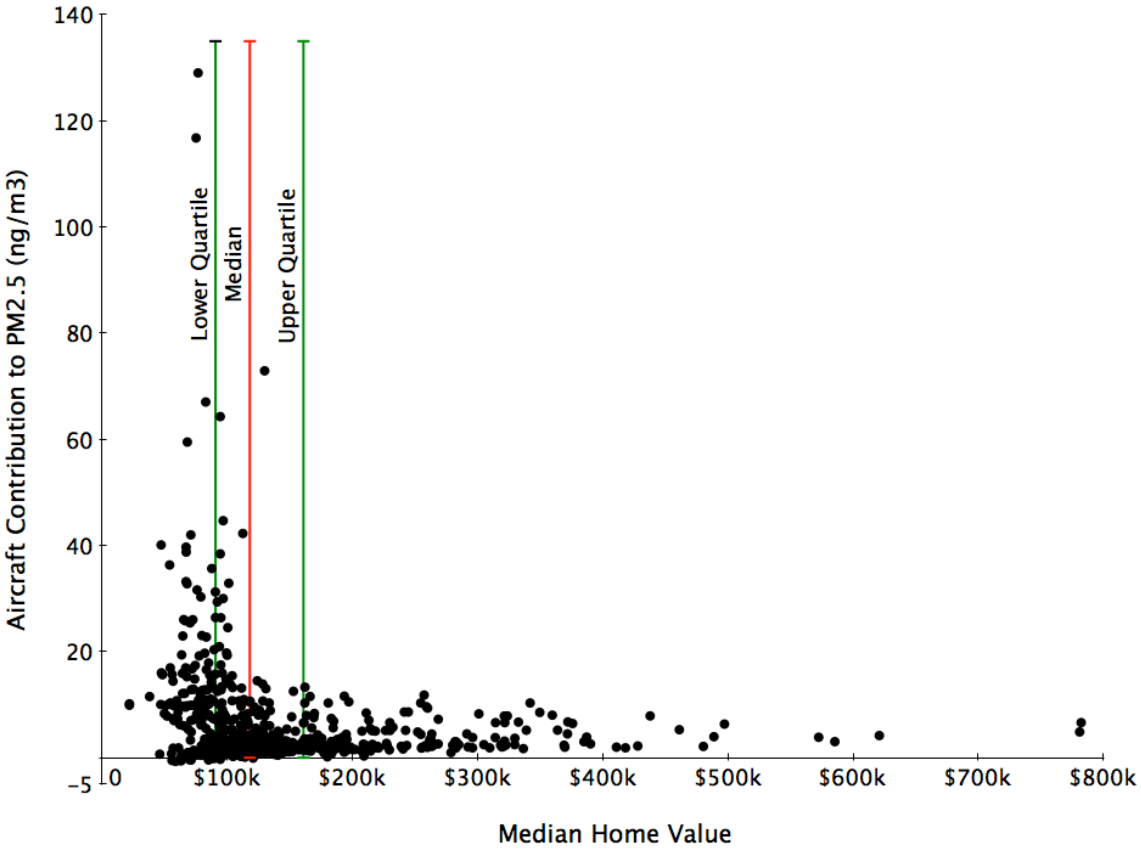

Figure 21 - Median home value vs. aircraft contribution to average $PM_{2.5}$ in June and July 2002

	Contrib From Aircraft			
Median Home Val	Under20ng	20to50ng	Over50ng	Total
LowestQuartile	152	19	4	175
SecondQuartile	166	10	1	177
ThirdQuartile	175	0	1	176
UpperQuartile	177	0	0	177
Total	670	29	6	705

Fisher's exact = 0.000

Table 8 - Median home value vs. aircraft contribution to average PM2.5 in June and July 2002 evaluated with Fisher's exact test for statistical significance

In summary, there appears to be no significant relationship between aircraft contribution to $PM_{2.5}$ and the studied demographic variables at low levels of aircraft-

contributed particulate matter pollution. However, tracts that suffer from high levels of particulate matter pollution are disproportionately low-income and nonwhite. They have lower median home values and higher percentages of adults who have not attained high school diplomas.

Exposure and Health

Finally, we wish to provide a rough estimate of mortality due to aircraft-contributed $PM_{2.5}$, which may help to provide a human context for the concentration numbers we have discussed. To make this estimate, we use a concentration-response function that relates changes in average $PM_{2.5}$ concentration to mortality. A number of studies have investigated this concentration-response function. First, note that a study by Schwartz et al. (2008) found that the relationship between risk of death and average $PM_{2.5}$ concentration is linear, and there is no evidence of any threshold below which exposure does not increase mortality risk in this way (Schwartz et al., 2008). Therefore, the concentration-response function we use can be appropriately applied to all census tracts irrespective of background $PM_{2.5}$ concentrations.

The relative risk (RR) of an exposure is the ratio of the risk of an adverse outcome for an exposed population to the risk of that adverse outcome in a non-exposed population. A study by Laden et al. (2006) examined deaths among a cohort of 8,096 adults (age 25+) from 1974-1998 in six U.S. cities where regular $PM_{2.5}$ measurements were taken (as part of the Harvard Six Cities study; Dockery et al., 1993). They found a RR ratio for overall mortality of 1.16 for each 10 $\mu g/m^3$ increase in average $PM_{2.5}$ concentrations for adults age 25+. In other words, there are 16% more adult deaths in a

population exposed to an additional 10 μg/m^3 average $PM_{2.5}$ concentration than there would have been in the absence of that elevated concentration. A study by Pope et al. (2002) examined data from an ongoing American Cancer Society study of 1.2 million adults (age 30+) from around the country (Chao et al., 2000). Pope et al. estimated the RR to be 1.04 per 10 μg/m^3 (i.e., an increase of 4% in all-cause mortality for each additional 10 μg/m^3 average $PM_{2.5}$). An analysis consolidating the views of 12 health experts who have conducted research on $PM_{2.5}$ exposure and mortality estimated a reduction of 1 μg/m^3 in annual average $PM_{2.5}$ concentration reduces the annual adult mortality rate by 0.7% – 1.6% (Industrial Economics, 2006).

We assume a 1% increase in all-cause adult mortality per 1 μg/m^3 average $PM_{2.5}$ contribution due to aircraft. This lies between the estimates from Laden et al. (2006) and Pope et al. (2002), and it is on the low end of the range found by the Industrial Economics study (2006).

We use the method from Anenberg et al. (2010) to calculate the number of premature deaths due to aircraft. RR and concentration are related via the log-linear relationship:

$$RR = \exp^{\beta * \Delta X} \qquad [2]$$

In this equation, RR is the relative risk for an exposure of a certain magnitude (in this case, 1.01) and ΔX is the change in concentration associated with that increase in risk (here, 1 μg/m^3). Accordingly, we are able to calculate the constant β=0.00995. The fraction of deaths attributable to aircraft, or AF, is defined as:

$$AF = \frac{RR - 1}{RR} = 1 - \exp^{-\beta * \Delta X} \quad [3]$$

AF values are specific to each county, so in Equation 3 ΔX is the aircraft contribution to a county's average $PM_{2.5}$ concentration. For each county, we multiply AF by the baseline adult mortality rate (the number of adult deaths (age 25+) in that county in 2002, or D_{2002}) to determine the change in mortality due to aircraft:

$$\Delta Mort = D_{2002} * (1 - \exp^{-\beta * \Delta X}) \quad [4]$$

The total number of adult deaths (age 25+) in 2002 for each county in the 29-county Atlanta area was obtained from the CDC WONDER database (CDC, 2010). Average $PM_{2.5}$ contribution due to aircraft was assigned to each county with a weighted average of grid cell pollutant values, similarly to the way in which concentrations were assigned to census tracts. Equation 4 was used to calculate the deaths attributable to aircraft for each county, and results were summed for all counties. Assuming that aircraft impact on average $PM_{2.5}$ concentrations in June and July 2002 is representative of their average impact on $PM_{2.5}$ concentrations throughout 2002, we find aircraft LTO emissions were responsible for 1.4 premature adult deaths in 2002.

Discussion and Conclusions

Our research shows measurable aircraft impacts on air quality across the 29-county region. Aircraft contributed less than 15 ng/m^3 to average $PM_{2.5}$ concentrations in most census tracts in the months of June and July, 2002. Below this concentration threshold, there was little relationship between aircraft-contributed $PM_{2.5}$ and any demographic variables examined.

However, there were 37 census tracts to which aircraft contributed more than 20 ng/m^3, and aircraft contribution was greater than 50 ng/m^3 in eight tracts. These census tracts overwhelmingly tended to have lower than average median household income and median house values. They also had above-average percentages of nonwhite population and population of age 25+ without a high school diploma.

Since aircraft only contribute a small fraction of total $PM_{2.5}$, aircraft contribution to air pollution levels cannot be readily observed or measured by ordinary residents.[1] Therefore, the strong observed correlation at high aircraft-contributed pollution levels may be mediated by another variable that is observable by residents. For instance, perhaps areas near the airport are dominated by warehousing, cargo handling, parking, inexpensive hotels, and other support services commonly considered to be unattractive land uses. Additionally, the airport has both noise (Hartsfield-Jackson, 2011) and traffic impacts. Wealthier residents may be willing to pay more to live farther away from these impacts. As a result, housing near the airport may be more affordable for low-income individuals.

Aircraft impacts on $PM_{2.5}$ extend over hundreds of kilometers, as shown in this study and elsewhere (Arunachalam et al., 2008, 2011). Therefore, it would be extremely difficult to completely eliminate human exposure to particulate matter from aircraft exhaust while siting airports in locations that are convenient for travelers and businesses. However, areas that experience a high $PM_{2.5}$ contribution from airports are more limited. It may be feasible for planners and local governments to reduce or eliminate exposure to

[1] Even a resident with precise measurement equipment would have trouble disentangling aircraft impacts from ambient urban $PM_{2.5}$ concentrations.

the highest concentrations of aircraft exhaust by focusing on these areas. For instance, cities may wish to consider discouraging development that would locate large numbers of people, or vulnerable populations such as the young, sick, or elderly, in areas likely to experience high levels of air pollution.

If resources allow, local governments near to existing or planned airports should consider conducting or funding an atmospheric modeling study to determine the actual or potential impact of aircraft emissions on their communities and the spatial scales of these impacts. (These studies should exceed the minimum FAA requirements.) Modeling studies have two important advantages over air quality measurements: the influence of aircraft can be distinguished from that of other emissions sources by comparing a base test case to one which omits aircraft emissions, and a model can be used to estimate aircraft impact under various scenarios, such as a planned airport expansion. For example, a future year scenario could be modeled, assuming changes in background emissions (due to new or improved control technologies, regulations, or changes in GDP) and differences in aircraft emissions (due to changes in fleet composition and air travel demand). Local governments should also be aware that atmospheric models have limitations, such as the fact that they often have error and bias relative to real world concentrations. Use of measurements in conjunction with modeling can help to overcome the shortcomings of each technique when used alone.

When conducting air quality modeling for planning purposes, it will be important for local governments to look at the absolute levels of modeled pollutant concentrations, not just the contribution from a particular source, as was the focus of this study. Results that assume different emissions scenarios (for instance, alternative traffic emissions, land

uses, or regulations affecting individual behavior) can each be modeled to determine the likely environmental consequences of various planning and policy options.

Additionally, since air pollution is largely invisible to residents, they may not have the information necessary to make effective decisions about where to live or to locate their businesses. By making the results of air quality modeling studies publicly accessible, residents may be better informed about the quantity and major emitters of air pollution in their community.

One important finding of this study is that populations which have less income, less education, lower home values, and are more nonwhite are more likely to be located in areas which are strongly affected by aircraft emissions. We do not address whether this co-location occurred before or after airport establishment, a question of cause-and-effect. Local governments should avoid placing public housing projects or directing affordable housing developments to these locations, since populations with less income and education are less likely to possess health insurance (DeNavas-Walt et al., 2010; Martinez et al., 2011) and therefore are less able to effectively address any health problems which may arise due to exposure to air pollution. If feasible, local governments should consider zoning impacted areas for low-density non-residential uses, such as warehousing, utilities, parkland, and possibly some types of manufacturing (Papsidero, 1992).

Future Work

There are a number of promising avenues available to future researchers, planners, and air quality modelers interested in this area of inquiry.

While this study examined average concentrations, which highlights the areas experiencing the most frequent and significant impacts, time-averages do not provide information about the magnitude of impacts that might be experienced by individuals at a particular time. An examination of hourly modeling data, such as that produced as part of our modeling study, could reveal the concentrations of aircraft-derived pollutants to which people are sometimes exposed.

Although this study estimated premature adult deaths due to aircraft contribution to $PM_{2.5}$ in 2002, this estimate is conservative because aircraft impacts on gas phase pollutants were not studied, emissions outside the landing and take off (LTO) cycle were not included, and our model results were biased low relative to monitor observations (see Chapter I of this thesis). Furthermore, we assumed that aircraft impacts in June and July 2002 are representative of the whole year, we did not estimate deaths among people of age 0-24, and we did not look at adverse health outcomes other than death (i.e. morbidity). A more thorough health impact analysis could take additional emissions into account (cruise-phase emissions and/or non-PM impacts) and model emissions for a longer time period. Premature deaths could be apportioned by census variables (race, income, etc.), and the indirect costs associated with those premature deaths could be calculated.

Investigators may wish to learn whether people living in tracts experiencing high levels of aircraft $PM_{2.5}$ contribution are aware of the aviation emissions, how long they have been living in the area, and why they decided to move there. This type of study could be based on interviews. It would provide a sociological perspective on this issue

and help to determine the nature of the cause-and-effect relationship between $PM_{2.5}$ contribution and low-resource communities.

Finally, advocacy planners or others interested in community involvement and environmental justice could work with affected populations to push for the adoption of technologies and regulations that limit aircraft emissions.

References for Chapter 2

Anenberg, S., Horowitz, L., Tong, D., West, J., 2010. An Estimate of the Global Burden of Anthropogenic Ozone and Fine Particulate Matter on Premature Human Mortality Using Atmospheric Modeling. *Environmental Health Perspectives* 11(9):1189-95.

Atlanta Regional Commission, 2011. Atlanta Region Information System (ARIS) Volume 1c. Georgia GIS Clearinghouse. http://data.georgiaspatial.org/index.asp?body=preview&dataId=42147. Accessed 2-6-11.

Arunachalam, S., B.H. Baek, A. Holland, Z. Adelman, F.S. Binkowski, A. Hanna, T. Thrasher and P. Soucacos, 2008. *An Improved Method to Represent Aviation Emissions in Air Quality Modeling Systems and their Impacts on Air Quality,* In Proceedings of the 13th Conference on Aviation, Range and Aerospace Meteorology, New Orleans, LA, Jan 2008.

Arunachalam, S., B Wang, N Davis, BH Baek, and JI Levy, 2011. Effect of Chemistry-Transport Model Scale and Resolution on 1 Population Exposure to $PM_{2.5}$ from Aircraft Emissions during Landing and Takeoff**,** *Atmospheric Environment*, doi:10.1016/j.atmosenv.2011.03.029.

Button, K., and Lall, S., 1999. The Economics of Being an Airport Hub City. *Research in Transportation Economics* 5, 75-105.

Centers for Disease Control and Prevention, 2010. National Center for Health Statistics. Compressed Mortality File 1999-2007. CDC WONDER On-line Database, compiled from Compressed Mortality File 1999-2007 Series 20 No. 2M, 2010. http://wonder.cdc.gov/cmf-icd10.html. Accessed 4-8-11.

Chao, A, Thun MJ, Jacobs E, Henley SJ, Rod- riguez C, Calle EE., 2000. Cigarette smoking and colorectal cancer mortality in the Cancer Prevention Study II. *J Natl Cancer Inst.* 92:1888-1896.

DeNavas-Walt, C., Proctor, B., Smith, J., 2010. Income, Poverty, and Health Insurance Coverage in the United States: 2009. U.S. Census Bureau. http://www.census.gov/prod/2010pubs/p60-238.pdf. Accessed 4-8-11.

Dockery DW, Pope CA III, Xu X, Spengler JD, Ware JH, Fay ME, Ferris BG Jr, Speizer FE., 1993. An association between air pollution and mortality in six US cities. *N Engl J Med* 329:1753–1759.

Federal Aviation Administration, 2011. Land Use Compatibility and Airports. http://www.faa.gov/about/office_org/headquarters_offices/apl/noise_emissions/planning_toolkit/media/III.B.pdf. Accessed 2-5-11.

Graham, A., 2008. Managing Airports: An International Perspective. Third Edition. Elsevier: Oxford, UK.

Hartsfield-Jackson Atlanta International Airport, 2009. Connecting the Region to the World: 2009 Economic Impact Study. http://www.atlanta-airport.com/docs/NewsRoom/2009_Economic_Impact_Study_report.pdf. Accessed 4-7-11.

Hartsfield-Jackson Atlanta International Airport, 2011. The H-JAIA Noise Insulation Program (NIP). http://www.atlanta-airport.com/Airport/NIP/default.aspx. Accessed 2-5-11.

Industrial Economics, 2006. Expanded Expert Judgment Assessment of the Concentration-Response Relationship between $PM_{2.5}$ Exposure and Mortality. Cambridge, MA: Prepared for the Office of Air Quality Planning and Standards, U.S. Environmental Protection Agency. http://www.epa.gov/ttn/ecas/regdata/Uncertainty/pm_ee_report.pdf. Accessed 4-7-11.

Karamchandani, P., Vijayaraghavan, K., Chen, S., Balmori-Bronson, R., Knipping, E., 2010. Development and application of a parallelized version of the advanced modeling system for transport, emissions, reactions and deposition of atmospheric matter (AMSTERDAM): 1. Model performance evaluation and impacts of plume–in–grid treatment. *Atmospheric Pollution Research* 1 260-270.

Lu, C., 2011. The economic benefits and environmental costs of airport operations: Taiwan Taoyuan International Airport, Journal of Air Transport Management, doi:10.1016/j.jairtraman.2011.02.006

Martinez, M., Cohen, R., 2011. Health Insurance Coverage: Early Release of Estimates From the National Health Interview Survey, January – September 2010. Centers for Disease Control and Prevention. http://www.cdc.gov/nchs/data/nhis/earlyrelease/insur201103.pdf. Accessed 4-8-11.

Papsidero, V., 1992. Airport Noise Regulations. American Planning Association Planning Advisory Service, Report Number 437.

Pope C, Burnett R, Thun M, Calle E, Krewski D, Ito K, Thurston G, 2002. Lung cancer, cardiopulmonary mortality, and long-term exposure to fine particulate air pollution. JAMA; 287(9):1132-1141.

Ratliff, G., Sequeira, C., Waitz, I., Ohsfeldt, M., Thrasher, T., Graham, M., Thompson, T., 2009. Aircraft Impacts on Local and Regional Air Quality in the United States. *Partnership for AiR Transportation Noise and Emissions Reduction*. http://web.mit.edu/aeroastro/partner/reports/proj15/proj15finalreport.pdf. Accessed 4-8-11.

Rice, C., 2003. Validation of Approach and Climbout Times-in-Mode for Aircraft Emissions Computation. *2003 Transportation Research Board Annual Meeting.* www.ltrc.lsu.edu/TRB_82/TRB2003-001446.pdf. Accessed 2-28-11.

Schwartz J, Coull B, Laden F, Ryan L., 2008. The effect of dose and timing of dose on the association between airborne particles and survival. *Environmental Health Perspectives* 116(1):64-69.

Swift, H., 2010. A Review of the Literature Related to Potential Health Effects of Aircraft Noise. *Partnership for AiR Transportation Noise and Emissions Reduction*. http://web.mit.edu/aeroastro/partner/reports/proj19/proj19-healtheffectnoise.pdf. Accessed 4-7-11.

Tsai, F., Sun, W., Chen, J., 2001. A Composite Modeling Study of Civil Aircraft Impacts on Ozone and Sulfate over the Taiwan Area. *Journal of Terrestrial, Atmospheric, and Oceanic Sciences*. V 12, N 1.

U.S. Census, 2000. SF1 and SF3 datasets. http://www.census.gov/.

U.S. Environmental Protection Agency. National Ambient Air Quality Standards (NAAQS). http://epa.gov/air/criteria.html. Accessed 4-8-11.

Waitz, I., Townsend, J., Cutcher-Gershenfeld, J., Greitzer, E., and Kerrebrock, J., 2004. Aviation and the Environment: A National Vision Statement, Framework for Goals and Recommended Actions. *Partnership for AiR Transportation Noise and Emissions Reduction*. http://web.mit.edu/aeroastro/partner/reports/congrept_aviation_envirn.pdf. Accessed 5-7-11.