PUFF – Volcanic Ash Dispersion Modeling
PUFF-AFWA Version 3.00
Functional Requirements Document

Document Number UPOS-B33-01

L. Nguyen

25 Sept 2003

Table of Contents

List of Figures and Tables

1. Introduction

1.1. Purpose

This document sets forth the functional requirements of the PUFF volcanic ash dispersion modeling programs, which predict the geographical distribution of volcanic ash from an eruption versus time. This document describes the requirements of the PUFF suite as a whole, the special requirements of the wind conversion program (afwa2puff) and the form and content of the inputs required by these executables and their resultant outputs.

The functional requirements given in this document do not necessarily represent the requirements that PUFF was originally meant to satisfy in its operational deployment in Alaska. Rather, they reflect the functionality present in the latest version, augmented by a small number of postulated requirements arising from the anticipated operational environment at the Air Force Weather Agency (AFWA).

The functional requirements will be used as a basis for the development and execution of test plans/procedures.

1.2. Background

PUFF is a volcanic ash dispersion prediction tool. PUFF was developed at the Geophysical Institute, University of Alaska Fairbanks and used by the Alaska Volcano Observatory (AVO) for volcano monitoring. Initially, PUFF was a research tool conceived by Dr. Hiroshi Tanaka for predicting the movement of eruption clouds. Dr. Craig Searcy conceived and developed the present version of PUFF as part of his PhD program. This version is used by the National Weather Service (NWS) and AVO to track volcanic eruption clouds.

Refinements in the Graphical User Interface (GUI) and data conversions were implemented by The Johns Hopkins University Applied Physics Laboratory (JHU/APL) in a joint project with the University of Alaska. JHU/APL is responsible for modifying the program and/or developing supporting utilities to facilitate its deployment at the Air Force Weather Agency (AFWA) site at Offutt AFB, NE. An additional responsibility is to develop a basic documentation set including this document. The system is currently in operation at the Air Force Weather Agency (AFWA).

1.3. Overview

The PUFF program models the dispersion of volcanic ash from an eruption and provides predictions of ash particle locations (latitude/longitude/altitude) versus time given eruption characteristics and wind field forecasts produced by another model. The PUFF application suite comprises five executable programs (puff, afwa2puff, puffgui, puffview, and ashdump) that provide the modeling capability; input data preprocessing; a graphical user interface (GUI) for model run specification; a GUI for viewing results; and utilities for viewing summaries of binary file contents. The application is written in C++, while the associated GUI functions are largely handled via the Tool Command Language (Tcl) scripts employing Toolkit (Tk) Motif widgets. The application suite can be hosted on Unix systems.

The PUFF model predicts the movement of ash particles ejected from a volcano versus time. The operator may select from a number of different initial conditions for the ash distribution and particle size. Particle locations are computed for each integration step (typically 5 minutes), with a snapshot of all particle locations at a given summation time (typically one or more hours) being written to an ash file.

For input, PUFF requires the name of a volcano, eruption characteristics, and forecasts of wind speeds for the time period of interest. The wind speed data must be available in gridded binary (GRIB) files. At AFWA, these required wind GRIB files are produced by a variety of models. PUFF's afwa2puff program converts the GRIB file outputs of the various wind models to U and V wind velocity versus geopotential height files. The U and V files serve as inputs for PUFF's volcanic ash tracking model. PUFF's puffview program displays a map of the area surrounding the volcano of interest overlaid with a graphical depiction of the ash distribution and overlaid with location identifier labels (pushpins). The PUFF volcanic ash tracking model outputs a series of ash files in Network Common Data (netCDF) format, describing the ash distribution over time.

The processes used in the model and an analysis of model results versus observations are given in “PUFF: A high-resolution volcanic ash tracking model,” (see reference 1).

1.4. Components

Table 1-1 identifies and provides a brief description of the roles of the PUFF application suite components.

Table 1-1 PUFF Application Suite Components

Executable	Role
puff	Contains the volcanic ash dispersion model and is executed for each model run.
puffgui	GUI invoked by the operator and used to specify model parameters and select source wind data for use by the model; automatically invokes afwa2puff (wind data conversion), puff (model), and puffview (view results) as necessary.
puffview	GUI normally invoked automatically by puffgui following a model run. This displays a map of an area surrounding the volcano of interest overlaid with a graphical depiction of the ash distribution and overlaid with location identifier labels (pushpins).
ashdump	Utility normally invoked by puffview to extract data from the ash files produced during the model run. It can also be invoked from the command line by a knowledgeable operator to inspect ash data.
afwa2puff	Utility normally invoked by puffgui to convert wind speed data contained in gridded binary (GRIB) files into a form usable by the puff executable. Can also be invoked from the command line to automate source wind file creation.

1.5. Document Organization

Section 1 describes the scope of the PUFF application system.
Section 2 lists applicable references.
Section 3 provides the detailed requirements for the PUFF program.
Section 4 provides the detailed requirements for the afwa2puff program.
Section 5 provides the requirements for the PUFF input and output data.
Section 6 provides a list of acronyms and abbreviations.

2. References

1. “PUFF: A high-resolution volcanic ash tracking model,” Journal of Volcanology and Geothermal Research 80 (1998) pp1-16, Craig Searcy, Ken Dean, and William Stringer

2. “PUFF User’s Guide for Puff version 2.5-AFWA-v110”, JHU/APL Memorandum SRS-99-198, 9 November 1999

3. “NetCDF User’s Guide for C – An Access Interface for Self-Describing, Portable Data”, Version 3, Russ Rew, Glenn Davis, Steve Emmerson, and Harvey Davies, Unidata Program Center, June 1997 - http://www.unidata.ucar.edu/packages/netcdf/index.html

4. “The WMO Format for the Storage of Weather Product Information and the Exchange of Weather Product Messages in Gridded Binary Form as used by NCEP Central Operations,” Clifford H. Dey, U.S. Department of Commerce, National Oceanic and Atmospheric Administration, National Weather Service, National Centers for Environmental Prediction Office Note 388, GRIB (Edition 1), 10 March 1998

3. PUFF Program

3.1. Operational Concept

3.1.1. Introduction

The operational concept described here is postulated based on limited information about AFWA's normal operational and does not represent an official operational concept.

3.1.2. Infrastructure

The PUFF program will reside on a Sun workstation. It shall be operable directly from that workstation, or from another network-connected workstation providing X-server capability.

Inputs

The program requires gridded U and V component wind speed data for the area of interest. These data shall be provided in Gridded Binary (GRIB) format on a latitude/longitude grid or on a Cartesian grid related to a polar stereographic, Mercator, or Lambert Conformal projection. These files may be resident on the provided workstation or on an accessible file system.

Outputs

The program will produce one or more files describing the ash distribution for each requested point in time (multiples of hour intervals).

3.1.3. Operation

The modeling process is initiated by typing a program name on the command line, with optional arguments. Nominally, parameters representing the initial conditions of the model are specified via a GUI that appears following entry of the command name.

Using the GUI, the user selects a volcano from a list of volcanos and specifies certain characteristics about the eruption and the simulation. If appropriate wind field data are not already available, the operator causes selected data to be extracted from the GRIB files and converted to netCDF format.

With model parameters set and windfield file data available, the operator invokes the PUFF model and files containing ash particle locations versus time are generated. The results contained in these files are viewed in a display window overlaid on a map projection. If desired, the images are printed or saved to a file. This process is repeated for each volcano of interest. The destination paths are changed between runs to prevent overwriting of result files.

3.2. Requirements

Following are the functional requirements for the PUFF application suite. The requirements are described in the following order: 1) Required functions, 2) Input data; 3) Processing; 4) Output data; and 5) Year 2000 operation.

1. FUNCTIONAL REQUIREMENTS

1.1 The program shall provide the capability to perform the following functions:

1.1.1 Predict ash distributions versus time given wind data for the area of interest and characteristics of the volcanic eruption.

1.1.1.1 Model Invocation

1.1.1.1.1 The application shall provide the operator the capability to invoke the modeling action via the following methods:

1.1.1.1.1.1 GUI

1.1.1.1.1.2 Command line

1.1.1.2 Modeling Options

1.1.1.2.1 The application shall provide the operator the capability to specify the following eruption-related characteristics:

1.1.1.2.1.1 Volcano name

1.1.1.2.1.2 Volcano regional location

1.1.1.2.1.3 Volcano latitude and longitude

1.1.1.2.1.4 Volcano cone elevation

1.1.1.2.1.5 Eruption date and time

1.1.1.2.1.6 Eruption duration

1.1.1.2.1.7 Minimum plume height (linear distribution only)

1.1.1.2.1.8 Maximum plume height

1.1.1.2.1.9 Vertical distribution of particles in plume

(a) Linear

(b) Poisson

(c) Exponential

1.1.1.2.1.10 Maximum horizontal extent of particles in plume at plume peak (defines an inverted cone with 0 horizontal extent at plume base)

1.1.1.2.1.11 Nominal vertical extent of particles below plume peak for Poisson and Exponential distributions

1.1.1.2.1.12 Initial particle count (number of ash particles)

1.1.1.2.1.13 Mean of a Gaussian distribution of particle sizes (in meters) on a logarithmic scale

1.1.1.2.1.14 Standard deviation of a Gaussian distribution of particle sizes on a logarithmic scale

1.1.1.2.1.15 Restart file name for multiple eruptions and existing ash clouds.

1.1.1.2.2 The application shall provide the operator the capability to specify the following simulation characteristics:

1.1.1.2.2.1 Simulation run time

1.1.1.2.2.2 Time intervals at which ash locations are saved to files

1.1.1.2.2.3 Integration time (time interval at which ash locations are computed)

1.1.1.2.2.4 Windfield data source

1.1.1.3 Modeling and Simulation Parameter Input Options

1.1.1.3.1 The application shall provide the capability to specify modeling options via the following methods:

1.1.1.3.1.1 GUI

1.1.1.3.1.2 Text file

1.1.1.3.1.3 Command line arguments

1.1.1.4 Windfield Data Input

1.1.1.4.1 The application shall be capable of obtaining U wind speed, V wind speed, and geopotential height versus pressure level data from records in GRIB format files based on the following grid types:

1.1.1.4.1.1 Latitude/longitude

1.1.1.4.1.2 Cartesian

1.1.1.4.2 For Cartesian gridded data, the program shall accept data described in terms of the following map projections:

1.1.1.4.2.1 Polar Stereo

1.1.1.4.2.2 Lambert Conformal

1.1.1.4.2.3 Mercator

1.1.1.5 Ash Location Data Output

1.1.1.5.1 The application shall record the location of ash particles at simulation time intervals specified by the save file interval.

1.1.2 Display the ash distribution predictions overlaid on a map, using color to differentiate particles at different altitudes.

1.1.2.1 View Options

1.1.2.1.1 The application shall provide the operator the capability to specify the following view options:

1.1.2.1.1.1 Map projections

(a) North Polar Stereo for areas entirely in the Northern Hemisphere

(b) South Polar Stereo for areas entirely in the Southern Hemisphere

(c) Mercator for areas between 85 deg North latitude and -85 degrees South latitude

1.1.2.1.1.2 Range of map display by specifying latitude/longitude points of the desired extent

1.1.2.1.1.3 Selection of one or more ash files to open

1.1.2.1.1.4 Selection of one or more open ash files to display

1.1.2.1.1.5 Animation of ash motion via sequential display of data from successive ash files

1.1.2.1.1.6 Association between ash symbol colors and ash elevation

(a) Choice of 16 predefined colors for any defined level

(b) Mapping of ash altitude to one of at least one, but no more than 16, distinct levels

(c) Altitude of highest ash displayed

(d) Altitude of lowest ash displayed

1.1.2.1.1.7 Selection of ash symbol size in range from no symbol (individual point) to 10 pixel symbols

1.1.2.1.1.8 Produce a Postscript file containing the map depiction and ash overlay and send it to a user-specified print queue or save the file to a user-specified filename.

1.1.2.1.1.9 Selection of ash symbol color tables.

2. INPUT DATA

2.1 Constraints on input data:

2.1.1 Eruption duration: specified in hours; must be <= simulation duration.

2.1.2 Minimum plume height: 0 <= minimum plume height < maximum plume height (integer – feet)

2.1.3 Maximum plume height: minimum plume height < maximum plume height <= 66,000 (integer – feet)

2.1.4 Initial particle count: 0 < count (integer)

2.1.5 Simulation run time: 1 < time <= 72 (real – hours)

2.1.6 Time interval at which ash files are saved: 1 <= interval <= simulation duration (integer – hours)

2.1.7 Integration interval: 1 <= interval <= 60 (integer – minutes)

2.2 Selection of source windfield data:

2.2.1 The program shall provide the operator with the capability to convert U wind speed, V wind speed, and geopotential height versus pressure level records from GRIB files into U wind speed and V wind speed versus geopotential height netCDF files with the following specifiable options:

2.2.1.1 Source wind model from the following choices: aviation (AVN), Navy Operational Global Atmospheric Prediction System (NOGAPS), Fifth Generation Mesoscale Model (MM5), AVN-datacube, and MRGF-datacube

2.2.1.2 Eruption time (earliest time for which wind data are required).

2.2.1.3 Total time coverage of interest. This can range from 1 hours to 72 hours based on the assumed availability of 72 hours of forecast data.

2.2.1.4 Minimum and desired numbers of hours of data to convert. The program shall consider it a success to find at least the minimum number of forecast hours but shall seek to satisfy the desired number if available. If at least the minimum number of hours are not yet available for a given model run, the function shall examine the next previous model run, until all available model runs are exhausted.

2.2.1.5 Relative latitude and longitude range (a single value for each) around the volcano (presumed to be the center point of the windfield data to be converted) for the case of global data sources.

2.2.1.6 Theater and nest of interest for the MM5 model data source

2.2.1.7 Path to the netCDF output file destination directory

2.2.2 The program shall distinguish GRIB source files for the various wind models by directory. For all cases except MM5, the files are contained in the "parent" directory for that model. For MM5, there are subdirectories in the parent directory for each defined theater. The files are stored in the theater directory corresponding to their geographical coverage.

2.2.3 The program shall maintain an association between default directory and source wind model and provide the capability for the user to override these defaults on the command line or via a text input file upon program invocation.

2.2.4 The program shall use the wind model output from the latest wind model cycle that covers the time period from the operator-specified eruption time to the operator-specified minimum number of hours hence.

2.2.5 The program shall support specification of forecast filename conventions for each source model, providing a simple method for specifying and changing them. These conventions can change, and such changes should not necessitate source code modifications.

2.2.6 In the GUI case, the program shall provide the capability to select the input windfield files for use by specifying the model and selecting a cycle (the time of the initial forecast) from a list of those available. For the MM5 case, the program shall list the theater and nest for each cycle.

2.2.7 In the GUI case, the program shall display information about the latitude/longitude/time coverage of a selected input windfield file.

3. PROCESSING

3.1 For each particle in the ash set the program shall calculate the total distance and direction of movement for each integration interval.

3.1.1 The program shall derive vertical wind velocities for each grid point present in the U and V windfield data.

3.1.2 The program shall apply the following processes (as described in Reference 1) to determine ash particle movement:

3.1.2.1 Transport – based on wind velocity at particle location

3.1.2.2 Turbulent dispersion – based on a random walk process

3.1.2.3 Fallout – based on particle size

3.1.3 The program shall terminate processing following completion of the specified analysis time or at the completion of the first integration step where a particle is determined to have left the area covered by the windfield data, whichever comes first.

4. OUTPUT DATA

4.1 The program shall save ash particle location data in a netCDF file format with longitude, latitude, geopotential height, size, and age data for each particle.

4.2 The program shall provide as an option the capability to save the derived vertical wind velocity file created during data preprocessing.

4.3 The program shall provide as an option the capability to save the initial ash data (ash distribution at start of simulation).

4.4 The program shall maintain a log during the model execution that contains summary information about the extent of the input windfield files and the values of modeling and simulation options in use for the run.

4.5 The program shall provide the capability to save an image containing the displayed map and ash distribution(s) as follows:

4.5.1 The program shall provide the following options for saved file format:

4.5.1.1 Postscript

4.5.1.2 Graphics Interchange Format (GIF)

4.5.1.3 Animated GIF (if two or more ash files are available)

4.5.2 The program shall include the following ancillary information in the image:

4.5.2.1 Ash height bar legend

4.5.2.2 Volcano name

4.5.2.3 Volcano latitude and longitude

4.5.2.4 Eruption date and time

4.5.2.5 Date and time image was created

4.5.2.6 Date and time of forecast for which this ash distribution is valid

4.5.3 The program shall provide default values in pixels for the width and height of GIF and animated GIF files and shall provide the capability for the operator to override the defaults on a case-by-case basis.

4.5.4 The program shall provide the capability to save a file describing the geometry and ash density of an existing ash cloud.

5. YEAR 2000 OPERATION

5.1 The program shall be capable of operating correctly for dates in the 21^{st} century.

4. AFWA2PUFF Program

4.1. Summary of Functions

The conversion program shall provide the following major functions:

1. Produce PUFF input files conforming to the format, organization, and content specified in section 5.2.2 from GRIB sources containing U and V wind speed data and geopotential height data versus pressure level.
2. Allow operator selection of the source wind model from the following choices: aviation (AVN), Navy Operational Global Atmospheric Prediction System (NOGAPS), Fifth Generation Mesoscale Model (MM5), AVN-datacube, and MRGF-datacube.
3. Allow operator specification of the forecast coverage time extent in hours up to and including 72 hours.
4. For global latitude/longitude based GRIB files, allow operator selectable latitude and longitude ranges bounding the point of interest for which wind speed data will be converted.
5. For MM5 data source, allow operator selection of theater and nest of interest.
6. Provide sufficient information in the resulting netCDF files for other processes to determine the time and area coverage of the files and define the grid upon which the data are projected.
7. Be operable from the command line or via a graphical user interface (GUI) integrated with the PUFF application front end GUI.

4.2. Requirements

Following are the detailed and derived requirements for the afwa2puff wind data conversion program. The requirements are described in the following order: 1) Required functions; 2) Input data; 3) Processing; and 4) Output data.

1. FUNCTIONAL REQUIREMENTS

   The conversion program shall provide the capability to perform the following functions:

   1.1 Produce PUFF input files conforming to the format, organization, and content specified in section 5.2.2 from GRIB sources containing U and V wind speed data and geopotential height data versus pressure level.

      1.1.1 Allow operator selection of the source wind model from the following choices:

1.1.1.1 aviation (AVN)

1.1.1.2 Navy Operational Global Atmospheric Prediction System (NOGAPS)

1.1.1.3 Fifth Generation Mesoscale Model (MM5)

1.1.1.4 AVN-datacube

1.1.1.5 MRGF-datacube

1.1.2 Allow operator specification of the forecast coverage time extent in hours up to and including 72 hours.

1.1.3 For global latitude/longitude based GRIB files, allow operator selectable latitude and longitude ranges bounding the point of interest for which wind speed data will be converted.

1.1.4 For MM5 data source, allow operator selection of theater and nest of interest.

1.1.5 Be operable from the command line or via a graphical user interface (GUI) integrated with the PUFF application front end GUI (puffgui).

1.1.6 For the case where MM5 data will be converted via the GUI interface, dynamically determine which theaters have been defined, and following operator selection of a theater from this set, dynamically determine and present to the operator nests available for selection.

1.1.7 Support specification of forecast filename conventions for each source model, providing a simple method for specifying and changing them. These conventions can change, and such changes should not necessitate source code modifications. Following are current descriptions:

1.1.7.1 AVN: Zero hour forecast/analysis data in "gblav.TxxZ.PGrbF00", where xx represents the cycle hours and can be 00 or 12 at this time. They may eventually add 06 and 18 model runs.

Subsequent forecast files are gblav.TxxZ.PGrbF03 – F78 and may not necessarily step in three-hour increments.

1.1.7.2 NOGAPS: Zero hour forecast/analysis data in "nogaps.TxxZ.anl", where xx represents the cycle hours and can be 00 or 12 at this time. They may eventually add 06 and 18 model runs.

The next forecast file in time sequence is nogaps.TxxZ.000 (1 hour forecast).

Subsequent forecast files are nogaps.TxxZ.003 – 096, and may not necessarily step in three-hour increments.

1.1.7.3 Datacube: (applies to both AVN and MRGF)
Zero hour forecast/analysis data in YYYYDOYHHMM0000, where YYYY is the 4-digit year, DOY is the three digit Julian date, HHMM are the hours and minutes of the cycle time, and 0000 represent the forecast hours.

Subsequent forecasts are YYYYDOYHHMM0000 – 0144.

1.1.7.4 MM5: Zero hour forecast in us057g1010txxyHHMMhhhmm, where xx represents the theater number, y represents the nest (only the "a" nest is guaranteed to have a 0 hour forecast), HHMM are the hours and minutes of the base cycle and can be anything, and hhhmm are the hours and minutes of the forecast, and will be all zeros for the base file in the series.

Subsequent forecast files will differ from the base in that the nest will reflect the nest desired by the PUFF operator and the hhhmm values will reflect the forecast time. These times do not have to be on even hours, and the interval between forecast files currently varies from one to three hours, but can be anything.

1.1.8 Alert the operator if it detects a wind model output file otherwise meeting usability criteria but having file permissions that prohibit its use by the program process. In such a case, the program shall continue looking for additional wind data for that run.

1.1.9 Convert wind speed data for all grid points in MM5 files.

1.1.10 Convert wind speed data for a user-specifiable subset of grid points centered on the volcano location for global latitude/longitude grids. The program shall provide the capability for the user to specify a symmetric range in latitude and longitude around the volcano location. If the delta latitude specification and volcano location are such that the latitude coverage extends over either pole, global longitude data must be converted for the requested range of latitudes.

1.1.11 Provide the capability for the user to specify minimum and desired numbers of hours of data to convert. The program shall consider it a success to find at least the minimum number of forecast hours but shall seek to satisfy the desired number if available. If at least the minimum number of hours are not yet available for a given model run, the function shall examine the next previous model run, until all available model runs are exhausted.

1.1.12 Provide a progress indication approximating the percentage complete (in terms of overall processing time).

1.1.13 Accept operator-specified values via command line arguments and via a text input file whose filename is provided as a command line argument.

2. INPUT DATA

2.1 Input parameter specification

2.1.1 Methods supported

2.1.2 The program shall support specification of input parameters as argument-value pairs on the command line, and as argument-value pairs given in an ASCII file whose filename is provided in an argument-value pair on the command line.

2.1.3 Parameters

2.1.3.1 Volcano latitude: Real number [-90.0, 90.0] degrees North

2.1.3.2 Volcano longitude: Real number [0.0, 360.0) degrees East

2.1.3.3 Maximum latitude delta: Real number [15.0, 180.0] degrees

2.1.3.4 Minimum latitude delta: Real number [-15.0, -180.0] degrees

2.1.3.5 Maximum longitude delta: Real number [30.0, 180.0] degrees

2.1.3.6 Minimum longitude delta: Real number [-30.0, -180.0] degrees

2.1.3.7 Minimum number of forecast hours to convert: Integer number [1, 72]

2.1.3.8 Desired number of forecast hours to convert: Integer number [1, 72]

2.1.3.9 Source wind model

2.1.3.10 Paths to parent directories for each wind model: Max of 180 characters each.

2.1.3.11 Directory in which to store resulting PUFF input files

2.1.3.12 Eruption time (UTC string) – Format "YYYY MM DD:HH"

2.1.3.13 Theater (MM5 model only)

2.1.3.14 Nest (MM5 model only)

2.1.4 Default values

The program shall support definition of default values for the following parameters:

2.1.4.1 Maximum latitude delta

2.1.4.2 Minimum latitude delta

2.1.4.3 Maximum longitude delta

2.1.4.4 Minimum longitude delta

2.1.4.5 Minimum number of forecast hours to convert

2.1.4.6 Desired number of forecast hours to convert

2.1.4.7 Source wind model

2.1.4.8 Paths to parent directories for each wind model

2.1.4.9 Directory in which to store resulting PUFF input files

2.2 Source wind file selection

2.2.1 Distinguish GRIB source files for the various wind models by directory as follows:

2.2.1.1 For all cases except MM5, the files are contained in the "parent" directory for that model.

2.2.1.2 For MM5, there are subdirectories in the parent directory for each defined theater. The files are stored in the theater directory corresponding to their geographical coverage.

2.2.1.3 Maintain an association between default directory and source wind model and provide the capability for the user to override these defaults on the command line or via a text input file upon program invocation.

2.2.2 Use the wind model output from the latest wind model cycle that covers the time period from the operator-specified eruption time to the operator-specified minimum number of hours hence.

2.2.3 Be capable of determining the applicable model cycle and the range of valid forecast times for all nests. There will always be an "a" nest file series for a given MM5 theater. The "a" nest always includes a zero hour forecast file (meaning applicable at the cycle time for the model run). The other nests (e.g., b, c, d) do not have to start, and usually do not start, with the model cycle time.

2.2.4 Select candidate source files within a directory as follows:

2.2.4.1 filenames match filename convention;

2.2.4.2 files were produced by a single model run whose cycle started at or before operator-specified eruption time;

2.2.4.3 sufficient forecast hour coverage exists to cover from the eruption time to eruption time plus and operator-specified minimum number of hours

2.2.5 Determine the order of storage of data in a GRIB file from the information in the GRIB Grid Description Section (GDS) header.

3. PROCESSING

The program shall be capable of performing the following processing steps:

3.1 Data selection

3.1.1 Sufficient source data determination

3.1.1.1 The program shall be capable of determining the range of source wind file data needed to satisfy a user's request given the user's specification of the minimum and desired number of hours to convert and the volcano eruption time.

3.1.1.2 The program shall notify the user if data for the minimum time coverage is not available.

3.1.2 Grid subset extraction

The program shall provide the capability to select a subset of the source wind file data points for cases where the source wind model data is projected on a global latitude/longitude grid as described in the following requirements:

3.1.2.1 Delta latitude specification

3.1.2.1.1 The program shall provide the capability to select only those source wind file data points corresponding to a latitude range defined by the volcano's latitude and the positive and negative delta latitude specified by the user.

3.1.2.1.2 In case either the Northerly or Southerly latitude extent was to extend over the pole, the program shall limit latitude extent to the pole and shall extend longitude coverage to complete coverage (i.e., 0-360 degrees longitude coverage).

3.1.2.2 Delta longitude specification

3.1.2.2.1 The program shall provide the capability to select only those source wind file data points corresponding to a longitude range defined by the volcano's longitude and the positive and negative delta longitude specified by the user.

3.2 Data ordering

3.2.1 The program shall be capable of determining from the GRIB header the ordering of data stored in a wind source file (any of four possible orders: increasing X, increasing Y; increasing X, decreasing Y; decreasing X, increasing Y; decreasing X, decreasing Y) and reordering it to the increasing X, increasing Y order required by PUFF.

3.3 Data conversion

3.3.1 The program shall be capable of converting wind speed data versus pressure level (in hPa) as provided by the GRIB files to wind speed data versus geopotential height (in meters) needed by PUFF.

4. OUTPUT DATA

4.1 The program shall produce two netCDF files containing wind speed values on a four-dimensional grid (forecast time, geopotential height, longitude/X, latitude/Y), one for U wind speed and one for V wind speed, as described in Section 5.2.2.

4.3. Operations Concept

PUFF requires source wind data and ancillary supporting data to be in netCDF files defined in a specific way (see section 5.2.2) and located in a default or user specified directory. Further, these files are expected to be named according to a specified convention.

The meteorological models run at AFWA produce data in GRIB format and store their files in directory locations specified for each model. These models are run multiple times a day, with each run requiring several hours to complete the generation of the collection of forecast files.

Because it would be costly (computationally and storage-wise) to produce PUFF input wind files for all available source model data, and the frequency of volcanic eruptions is low, the operational concept is for wind data to be converted on an as needed basis. For a given volcanic eruption, the user makes a decision as to which source model best handles modeling in that geographical region, and invokes the PUFF input file production process via a GUI. The user specifies a volcano (and thus the latitude and longitude around which the prediction will be centered), an eruption time, and the minimum and desired number of hours of wind data to convert. The user also specifies the source wind model, and in the case of MM5-produced data, the theater and nest of interest. The GUI invokes the afwa2puff program with these and other default parameters and the new PUFF input files are created. Once created, these input files can be selected from the “puffgui” GUI used to initialize and run the PUFF program.

If desired, afwa2puff can also be invoked in an automated fashion via a cron job. To support this mode of operation, different program invocations need to be specified for each source wind model type, and in the MM5 case, theater and nest combination. As mentioned earlier, however, this is not expected to be the normal mode of operation.

4.4. Constraints

The afwa2puff program operates under the following constraints/assumptions:

1. The source GRIB files’ Product Definition Sections (PDS) must be populated with accurate data.
2. The GDS must be included in all source GRIB files.
3. The grid must be fully populated (sparse grids are not supported).

4. For lat/lon based grids, the order of data storage in longitude must be from West to East (e.g., 0 degrees East, 1 degree East). This restriction can be easily removed, but is present in the version at the time of this writing. There is no restriction on latitude (North to South, or South to North), and there is no restriction in either dimension on X-Y grids.
5. Forecast files must be on hour increments (i.e., fractional forecast hours are not supported)

4.5. Validation and Testing

Testing and validation of the afwa2puff program will encompass verifying that the wind and ancillary data present in the resulting files agree with the source data from the GRIB files; the format of the resultant netCDF files conforms to that specified in section 5.2.2; the operational concept is fully supported; and the program handles the most common configuration and operational errors, providing informative feedback on the problem.

4.6. Possible Future Enhancements

Those items as defined in the "PUFF II" implementation under UPOS task BIX.

5. PUFF Data

5.1. Application Suite Argument Processing

The PUFF application suite uses a common mechanism for specifying options and their defaults. This mechanism supports the definition of independent default values for each of the executables, along with a consistent command line and text file method for specifying non-defaulted values or overriding default values.

Supported options are defined in *.args files. These text files comprise a line per argument containing the name, type, default value, and explanatory text for each argument (option). Each line is enclosed by a macro name (ARG) and parentheses. An excerpt from the puff.args file is shown in Figure 5-1.

```
ARG(saveHours,double,6,Save ash output at every saveHours interval)
ARG(dtMins,double,5,Step time interval in minutes)
ARG(fileU,string,"",Explicitly read U-component wind data file)
ARG(fileV,string,"",Explicitly read V-component wind data file)
```

Figure 5-1 Sample Default Argument Specification

There are several points worth noting:

- For an executable to be aware of an option, it must be defined in the *.args file associated with that executable in the Makefile
- The definition must be enclosed by `ARG( )`
- The option name must not contain spaces
- Valid option types are `string`, `double`, `long`, and `name`
- Each option should include an explanatory text string
- The executable must be completely rebuilt to support the option

Once an option is defined, it may be invoked using a command line flag or as a text entry in an options input file. If the latter form is used, the name of the file is provided on the command line using the `–argFile` option. The format for the information in the file is:

```
\<optName> value
```

For example:

```
\saveHours 6
```

The format for specification on the command line is:

```
-<optName> value
```

For example:

```
puff –argFile “myfile.txt”
```

5.2. PUFF Executable

5.2.1. User Data (Input)

The puff executable uses a subset of the options defined in puff.args. The puffgui and afwa2puff executables also use this file and are responsible for the presence of additional options not used by puff. Although each of the executables could have a separately defined argument file, there would be significant overlap with puff.args and thus changes to common elements would have to be made in multiple files. Table 5-1 lists the options from the puff.args file that are directly used by puff.

Table 5-1 User Specified Options used by the "puff" Executable

Option	Description	Default
volc	Volcano name	"unknown"
volcLon	Volcano longitude (decimal degrees East)	-9999
volcLat	Volcano latitude (decimal degrees North)	-9999
eruptDate	UTC Eruption date string in format "YYYY MM DD HH:MM"	""
runHours	Simulation length (hours)	24
saveHours	Save ash output at every saveHours interval (hours)	6
eruptHours	Continuously emit ash particles over this duration (hours)	24
dtMins	Step time interval (minutes)	5
fileU	Name of file containing U wind data	""
fileV	Name of file containing V wind data	""
path	Puff input wind file directory path	"."
nAsh	Number of ash particles initialized and tracked	3000
plumeMax	Maximum plume height in meters when plumeShape = linear	16000
plumeMin	Minimum plume height in meters when plumeShape = linear	0
plumeShape	Distribution of initial ash column in vertical – (l)inear; (p)oisson; (e)xponential	"linear"
plumeZwidth	Initialize ash vertical spread in meters	3000
plumeHwidth	Initialize ash horizontal spread in meters	0
diffuseH	Horizontal diffusion coefficient (m^2/integration period)	20000
diffuseZ	Vertical diffusion coefficient (m^2/integration period)	10
AshLogMean[1]	(See explanation following table)	-5
AshLogSdev[1]	(See explanation following table)	1
saveAshInit	Save the initial ash state if true	0 (false)
runSurface	Continue tracking particles at z=0 if true	0 (false)
noFallout	Do not include fallout in the ash particle motion	0 (false)
saveWfile	Save the divergence-created vertical wind to a file	0 (false)
argFile	Name of file from which to read optional arguments	""
quiet	Do not print out time value after each time step	0 (false)

Table 5-1 User Specified Options used by the "puff" Executable (cont'd)

Option	Description	Default
newline	Write newline after each time step	0 (false)
verbose	Make verbose output	0 (false)
help	Print this summary	0 (false)
volcPath	Volcano list filename	"/project/upos/puff/ puff-2.5- afwa110/src/volcan os.txt"
restartFile	Filename of either an existing ash file from a previous puff run (for multiple eruption) or filename of a cloud file describing an exsiting ash cloud (for run with existing cloud).	""
showVolcs	Display a list of predefined volcano sites and exit.	0 (false)
seed	Initial seed value for random processes.	random

Notes:

(1) The ash particle size distribution is gaussian on a *logarithmic* scale. That is, by default, the peak of the gaussian curve is at –5, representing an ash particle radius of 10^{-5} meters. The standard deviation *on the logarithmic scale* is 1, meaning that 60% of the particles will be between –4 and –6 on the scale, corresponding to sizes between 10^{-4} and 10^{-6} meters.

5.2.2. Wind Data (Input)

The puff executable requires wind speed input data in network Common Data Form (netCDF) format (Reference [3]). Two input wind files are required– one each for the U and V wind speed data, and the data must be presented on a grid. U and V are the vector components nominally along the horizontal and vertical axes of the grid, respectively. In AFWA's case, these data are produced by weather models and reside in GRIB format files. The afwa2puff executable is used to extract data from those files and produce the files required by puff (described here).

netCDF file contents are defined and populated by PUFF application suite components using the netCDF version 3.4 libraries. Version 3, or later, is required due to the use of certain data types. Additional information can be found on netCDF at http://www.unidata.ucar.edu/packages/netcdf/.

The netCDF libraries allow the definition of dimensions, variables, and attributes. Variables are considered to be arrays, with a scalar being a 0-length array. For puff, the wind data is stored as an array of values on a four-dimensional grid. The grid's dimensions are forecast time, level, row, and column for the Cartesian grid, and forecast time, level, latitude, and longitude for the latitude/longitude grid. The grid type, which is essential for the correct interpretation of the data, is included as a scalar value. Additional scalar data are provided to describe the grid upon which these data are projected.

Four dimensions are defined for use with the "data" array (rank 4) containing the actual wind data projected on the four dimensional grid. Four arrays (rank 1) are defined to store the values of the points along a particular dimension. For example, in Figure 5-2 frtime is defined as a dimension, with length 9, and as an array variable with "frtime" (9) entries. Each of these entries corresponds to the forecast time value for each point along that dimension. That dimension represents the number of hours after the model cycle time (reftime) at which the data are valid. Given that several of the wind models produce forecast files on 3-hour intervals, frtime contents for a file containing 24 hours worth of wind data might be { 0, 3, 6, 9, 12, 15, 18, 21, 24 }.

The values for the dimension arrays must increase as you move from the beginning of the array to the end. Increase in terms of latitude means moving from more southerly latitudes to more northerly latitudes. Increase in terms of longitude means moving from more westerly longitudes to more easterly longitudes. For example:

```
lat  = ( -40, -39, -38, …, 38, 39, 40)
lon = ( 75, 76, 77, 78, …, 220, 221, 222 )
lon = ( 358, 359, 0, 1, 2 ) // Meridian crossing case
```

The "data" array contains wind speed values corresponding to the four-tuples (frtime, level, row/lat, col/lon).

In addition to these data, each netCDF file contains a number of scalar values that contain data relating to the areal coverage of the file and the grid projection from which the data were derived.

Content summaries from two sample files, one based on a lat/lon grid and one based on a Mercator grid are given in Figures 5-2 and 5-3 respectively.

The numbers shown next to the dimensions correspond to the lengths (number of values) associated with those dimensions. For example, in the first example (the avn-based file), there are 9 forecast times, 26 geopotential levels, 31 latitude values, and 61 longitude values.

Latitude1/Longitude1 represents the point in the lower left corner of the grid, and the progression is clockwise. Consequently, Latitude2/Longitude2 is the upper left corner. These values are present to aid in the determination of adequate coverage when the operator selects input wind files in the PUFF GUI.

```
netcdf avn-003_1999040600_puffU {
dimensions:
      frtime = 9 ;  ß  Forecast times (9 times covered in file)
      level = 26 ;  ß  Geopotential height levels (26 in file)
      lat = 31 ;    ß  Number of latitude points in grid
      lon = 61 ;    ß  Number of longitude points in grid
      timelen = 20 ;  ß  Number of characters in reftime string
variables:
      float data(frtime, level, lat, lon) ;ß  Array of wind speed values
            data:_FillValue = -9999.f ;ß  Value to use if pt. is missing
            data:valid_range = -200.f, 200.f ;
            data:units = “m/sec” ;
      float frtime(frtime) ;ß  Array of forecast times (hours after
                               start of cycle) – Note that “frtime” is
                               defined both as a dimension (with a
                               length of 9), and as a variable array;
                               netCDF is able to distinguish between
                               these definitions
            frtime:units = “hours” ;
      float level(level) ;
            level:units = “geopotential meters” ;
      float lat(lat) ;
            lat:units = “degrees_north” ;
      float lon(lon) ;
            lon:units = “degrees_east” ;
      char reftime(timelen) ;ß  Time of original wind model run (cycle
                                time) in “YYYY MM DD HH:MM” format
      float latitude1 ;ß  latitude1 [degrees North] and longitude1
                          [degrees East] define the lower left corner of
                          the coverage zone (with respect to Earth with
                          North Pole at top)
      float latitude2 ;ß  latitude2, longitude2 define the upper left
                          corner
      float latitude3 ;ß  latitude3, longitude3 define the upper right
                          corner
      float latitude4 ;ß  latitude4, longitude4 define the lower right
                          corner
      float longitude1 ;
      float longitude2 ;
      float longitude3 ;
      float longitude4 ;
      int pds_grid_number ;ß  Grid number from the Product Definition
                              Section of the source GRIB file
      int gds_grid_number ;ß  Grid number from the Grid Description
                              Section of the source GRIB file
      char map_projection ;ß  G(lobal lat/lon), M(ercator), P(olar
                              Stereo), L(ambert Conformal)
      float delta_lat ;ß  Distance in degrees between adjacent latitude
                          points in the grid
      float delta_lon ;ß  Distance in degrees between adjacent longitude
                          points in the grid
```

Figure 5-2 Wind File Content Description for Lat/Lon Grid-Based Wind Data

```
      int rows ;ß  Number of "rows" of data in the grid (different
                   latitude values)
      int columns ;ß  Number of "columns" of data in the grid (different
                   longitude values)
// global attributes:
            :title = "PUFF U windfield data" ;
}
```

Figure 5-2 Wind File Content Description for Lat/Lon Grid-Based Wind Data (cont'd)

```
netcdf mm5t05a-255_1999110800_puffU {
dimensions:
      frtime = 2 ;
      level = 23 ;
      row = 176 ;
      col = 209 ;
      timelen = 20 ;
variables:
      float data(frtime, level, row, col) ;
            data:_FillValue = -9999.f ;
            data:valid_range = -200.f, 200.f ;
            data:units = "m/sec" ;
      float frtime(frtime) ;
            frtime:units = "hours" ;
      float level(level) ;
            level:units = "geopotential meters" ;
      float row(row) ;
            row:units = "km" ;
      float col(col) ;
            col:units = "km" ;
      char reftime(timelen) ;
      float latitude1 ;
      float latitude2 ;
      float latitude3 ;
      float latitude4 ;
      float longitude1 ;
      float longitude2 ;
      float longitude3 ;
      float longitude4 ;
      int pds_grid_number ;
      int gds_grid_number ;
      float Latin ;ß  From the GRIB GDS – needed for projection
                      calculations
      char map_projection ;
      float delta_row ;ß  Distance between adjacent rows (km)
      float delta_column ;ß  Distance between adjacent cols (km)
      int rows ;
      int columns ;

// global attributes:
            :title = "PUFF U windfield data" ;
}
```

Figure 5-3 Wind File Content Description for Mercator Grid-Based Wind Data

Some of the scalar values contained in the wind netCDF files are taken from the GRIB Grid Description Section (GDS) and are used to support calculations associated with the map projection. The set of required values varies by projection type. The Mercator projection case, shown in Figure 5-3, has one of these values (Latin). Figures 5-4 and 5-5 show the entries needed for Polar Stereo and Lambert Conformal cases (in place of the Latin entry for Mercator). Note that these figures show *only* the special projection-related quantities and not the remainder of the file data (since it will be the same as the Mercator case). These terms are defined in Table D of the GRIB Office Note (Reference [4]) and repeated in Figure 5-6 for convenience.

```
…
float vertical_longitude ;
char pole ;
…
```

Figure 5-4 Partial Listing of Wind File for Polar Stereo Case

```
…
float Latin1 ;
float Latin2 ;
float LatSP ;
float LonSP ;
char pole ;ß  'N' or 'S'
float vertical_longitude ;
…
```

Figure 5-5 Partial Listing of Wind File for Lambert Conformal Case

Vertical Longitude:	East longitude value of the meridian which is parallel to the y-axis (or columns of the grid) along which latitude increases as the y-coordinate increases.
Latin:	The latitude(s) at which the Mercator projection cylinder intersects the earth.
Latin1:	The first latitude from the pole at which the secant cone cuts the spherical earth.
Latin2:	The second latitude from the pole at which the secant cone cuts the spherical earth.
LatSP:	Latitude of southern pole.
LonSP:	Longitude of southern pole.

Figure 5-6 Definitions of Selected Wind File Projection-Related Scalar Values

The file naming convention for AFWA-generated files is:

<model>-<PDS grid number>_YYYYMMDDHH_puffU.cdf
<model>-<PDS grid number>_YYYYMMDDHH_puffV.cdf,

where <model> is one of {avn, nogaps, avn-cube, merge-cube, mm5}, and the date information refers to the wind model cycle time. For the case of MM5, there is additional theater and nest information (e.g., mm5t02a) included in the <model> entry. The PDS grid number must be three digits (e.g., 003, 255).

5.2.3. Ash Data (Output)

The puff executable produces one or more netCDF files containing ash information. An example of a single file is given in Figure 5-7. For each ash point, latitude, longitude, height, size, and age are included.

```
netcdf 99.096.0100_ash {
dimensions:
        nash = 3000 ;ß  Number of ash points in file

variables:
        double lon(nash) ;
                lon:units = "degrees_E" ;
        double lat(nash) ;
                lat:units = "degrees_N" ;
        double hgt(nash) ;ß  Geopotential height
                hgt:units = "meters" ;
        float size(nash) ;
                size:units = "meters" ;
        long age(nash) ;
                age:units = "time_t (seconds)" ;
        long clock_time ;
        long origin_time ;
        float origin_lon ;
        float origin_lat ;
        float erupt_hours ;
        float plume_height ;ß  Definition in Table 5-1 (plumeHeight),
                               but stored in meters
        float plume_width_z ;ß  Definition in Table 5-1 (plumeZwidth),
                                but stored in meters
        float plume_width_h ;ß  Definition in Table 5-1 (plumeHwidth),
                                but stored in meters
        float diffuse_h ;ß  Definition in Table 5-1
        float diffuse_v ;ß  Definition in Table 5-1
        float log_mean ;ß  Definition in Table 5-1
        float log_sdev ;ß  Definition in Table 5-1

// global attributes:
                :title = "Puff Ash Data" ;
                :volcano = "SHISHALDIN" ;
                :file_date = "99040600" ;
                :plume_shape = "linear" ;
}
```

Figure 5-7 Sample Ash Output File

Ash age, clock time, and origin time are reported in seconds since the Unix epoch (1/1/1970). Ash age is a bit of a misnomer, since it is actually the time (Unix seconds) at which the ash particle is included in the modeling. (This is absolute time, not time relative to the eruption or the simulation time at which the ash snapshot was saved to the file.) Origin time reflects the eruption time, and thus the time of earliest ash. Clock time represents the time at which this ash location "snapshot" was taken.

Given the number of ash particles to model, and the user-specified eruption time, PUFF distributes the "birth" times of the ash particles uniformly over the eruption time. For cases where ash files are produced for times prior to the end of the eruption, some of the ash particles in the file will have ages that are less than the clock time. For example, if the user specifies an eruption to last 6 hours, and chooses a Save Interval of 3 hours, some of the “age” values of ash particles in the first ash file will be smaller than that of the clock time for that file.

The naming convention used for the file is:

(a)aa.DOY.HHMM_ash.cdf

where (a)aa represents the number of years since 1900, DOY is the Julian day, and HHMM are the UTC hours and minutes corresponding to the time this ash location prediction is valid.

Ash files are stored in the working directory (i.e., the directory from which the puff executable was run).

5.3. PUFFGUI Executable

5.3.1. User Data (Input)

The puffgui executable serves as a GUI wrapper around the wind file conversion, puff model, and results viewing processes. Its role in life is to provide a method for an operator to specify modeling and wind file conversion-related parameters and then provide those parameters to the wind file conversion or puff model executable via a text file. In this way, it may indirectly use all of the arguments in puff.args although it does no significant processing itself.

The text files used for input to the afwa2puff file conversion (afwa2puff.input) and puff model executables (puff.input) include all parameters whose values have been changed by the operator from default values, plus selected other parameters. Some parameters, such as paths to GRIB parent directories, can be changed from their default values by specifying new values on the command line when invoking puffgui, but cannot be set through the GUI interface. Any parameter set on the command line when invoking puffgui is automatically included in the text input files generated for the afwa2puff and puff executables. Table 5-2, plus the contents of Table 5-1, comprise the set of parameters that can be set via the command line or via a text input file when starting puffgui.

Table 5-2 Additional Parameters Specifiable when invoking puffgui

Option	Description
model	Wind model for source (avn, nogaps, mm5, avn-datacube, merge-datacube)
avnPath[1]	Pathname to the AVN model GRIB file directory
nogapsPath[1]	Pathname to the NOGAPS model GRIB file directory
mm5Path[1]	Pathname to the MM5 model GRIB file directory
avnDatacubePath[1]	Pathname to the AVN-based Datacube GRIB file directory
mergeDatacubePath[1]	Pathname to the AVN and merged MM5-based Datacube GRIB file directory
lonDeltaMax[1,2]	Extent (from the volcano location) in the positive direction of East longitude for which wind data will be converted (positive number of degrees)
lon DeltaMin[1,2]	Extent (from the volcano location) in the negative direction of East longitude for which wind data will be converted (NEGATIVE number of degrees)
latDeltaMax[1,2]	Extent (from the volcano location) in the positive direction of North latitude for which wind data will be converted (positive number of degrees)
latDeltaMin[1,2]	Extent (from the volcano location) in the negative direction of North latitude for which wind data will be converted (NEGATIVE number of degrees)
theater[3]	MM5 theater of interest (integer; e.g., 3)
nest[3]	MM5 nest of interest (single char; e.g. “c”)
minConvertHours	Minimum number of hours of forecasts to convert (1 – 72)
desiredConvertHours	Desired number of hours of forecasts to convert (1 – 72)

Notes:
(1) These values need to be specified in the file ONLY if you want to use a non-default value. Additionally, even in that case the only GRIB file path that must be specified is the one for the model type you’re using (e.g., nogapsPath when using the NOGAPS model)
(2) Only for lat/lon based grids (e.g., NOGAPS/AVN/Datacube)
(3) Only for XY based grids (e.g., MM5)

5.3.2. Volcano Data (Input)

The puffgui executable reads the contents of a text file (volcanos.txt) located in the directory specified by the volcPath argument upon startup. It uses these data to populate arrays containing volcano names, locales, cone elevations, latitudes, and longitudes. These support the user selection of a volcano and the automatic population of related data. Information in this file was obtained from the Smithsonian Institution's Global Volcanism Program (http://www.volcano.si.edu/gvp/).

The file comprises volcano data records, one volcano per line in the following format:

```
volcano name:locale:latitude:latitude direction:longitude:longitude
direction:cone elevation
```

Volcano name:	Volcano name
Locale:	General geographic area for the volcano
Latitude:	Volcano latitude (degrees)
Latitude Direction:	N or S for North or South
Longitude:	Volcano longitude (degrees)
Longitude Direction:	E or W for East or West
Cone Elevation:	in meters (-99999 means unknown)

5.3.3. Parameter Data (Output)

The puffgui executable produces a parameter file for the wind file conversion process if the user invokes that function, and a parameter file for the puff model execution if the user invokes that function. Both follow the format described in Section 5.1 for text files containing argument parameters. For the sake of convenience, puffgui populates these files according to a single set of rules. The files will contain entries for all values that the user has specified via the GUI or the command line (potentially overriding the default values) plus possible additional entries containing the following information (regardless of whether the values are different than the defaults):

- Latitude Delta Minimum
- Latitude Delta Maximum
- Longitude Delta Minimum
- Longitude Delta Maximum
- AVN path
- NOGAPS path
- MM5 path
- AVN-datacube path
- Merge-datacube path
- Minimum convert hours
- Desired convert hours

These values are identified for inclusion in text input files if the user selects the GRIB convert … menu option during the session. The values have meaning only to the afwa2puff executable; they are ignored by the puff executable.

5.4. PUFFVIEW Executable

5.4.1. User Data (Input)

The puffview program provides a method for viewing the results of the puff model run. It creates a map image which is overlaid with the ash distribution using colored symbols and is overlaid with location identifiers (pushpins) consisting of a colored icon next to a location name. The puffview executable behaves slightly differently with respect to command line options, as it will accept one or more ash file names on the command line and open these upon initialization. Puffview uses its own arguments file (puffview.args) which has a minimal set of default parameters. These consist of the application name, version number, point of contact, and “help” option that causes this information to be listed.

Ash file names may be provided on the command line and will be opened and loaded by the program on startup. If multiple file names are provided, the set must be enclosed in quotes and there must be at least one blank space between adjacent names.

5.4.2. Ash Data (Input)

The puffview program uses as input the ash files described in Section 5.2.3.

5.5. ASHDUMP Executable

5.5.1. Ash Data (Input)

The ashdump program uses as input the ash files described in Section 5.2.3. It produces summaries of the file’s contents and writes them to standard output.

5.6. AFWA2PUFF Executable

5.6.1. User Data (Input)

The afwa2puff program accepts operator specification of the parameters given in Table 5-3 (a subset of the parameters from the puff.args file) using the methods outlined in Section 5.1.

Table 5-3 AFWA2PUFF Operator-Specifiable Options

Option	Description
model	Wind model for source (avn, nogaps, mm5, avn-datacube, merge-datacube)
avnPath	Pathname to the AVN model GRIB file directory
nogapsPath	Pathname to the NOGAPS model GRIB file directory
mm5Path	Pathname to the MM5 model GRIB file directory
avnDatacubePath	Pathname to the AVN-based Datacube GRIB file directory
mergeDatacubePath	Pathname to the AVN and merged MM5-based Datacube GRIB file directory
path	Pathname to directory where resultant windfield netCDF files will be written.
volcLat	Latitude (decimal degrees N) of the volcano of interest (the grid of data will be centered around this)
volcLon	Longitude (decimal degrees E) of the volcano of interest
lonDeltaMax	Extent (from the volcano location) in the positive direction of East longitude for which wind data will be converted (positive number of degrees)
lon DeltaMin	Extent (from the volcano location) in the negative direction of East longitude for which wind data will be converted (NEGATIVE number of degrees)
latDeltaMax	Extent (from the volcano location) in the positive direction of North latitude for which wind data will be converted (positive number of degrees)
latDeltaMin	Extent (from the volcano location) in the negative direction of North latitude for which wind data will be converted (NEGATIVE number of degrees)
theater	MM5 theater of interest (integer; e.g., 3)
nest	MM5 nest of interest (single char; e.g. “c”)
minConvertHours	Minimum number of hours of forecasts to convert (1 – 72)
desiredConvertHours	Desired number of hours of forecasts to convert (1 – 72)
eruptDate	Date/time string of volcano eruption of interest

5.6.2. Wind Data (Input)

The afwa2puff program extracts wind and geopotential height versus pressure level data from GRIB files, interpolates the wind data onto geopotential height levels, and writes the data to a netCDF file, as described in Section 5.2.2. The source wind files must be in GRIB Edition 1 form with fully populated Grid Description Sections. The following additional requirements hold:

The grid upon which the data are projected must be complete (i.e., thinned grids are not supported).

The projection must be one of: Global latitude/longitude, Polar Stereo, Lambert Conformal, or Mercator.

The file must contain U and V wind data versus pressure level in the UGRD and VGRD record types respectively.

The file must contain geopotential height data versus pressure level in the HGT record type.

Cartesian grids must have uniform spacing between points in the X and Y directions. That is, if the distance between adjacent X points is 100 km, the distance between adjacent Y points must be 100 km. This is a limitation of the software libraries used to convert between Cartesian and latitude/longitude values.

The program is not sensitive to the ordering of the records in the file, either in a given file, or on a file-to-file basis. Since each GRIB file contains data for a single forecast time, data are typically extracted from multiple GRIB files to produce the PUFF input wind files.

The program uses version 1.6 of the “wgrib” utility (http://wesley.wwb.noaa.gov/wgrib.html), as modified by AFWA to extract data and grid information. The program parses the textual grid description produced by wgrib and is thus dependent on the format of that description. This can change between versions of wgrib, and minor changes to the parsing code in projectionGrid.C will likely be required if a different wgrib version is used.

5.6.3. Wind Data (Output)

Afwa2puff produces netCDF files containing wind data as described in Section 5.2.2.

6. APPENDIXES

APPENDIX A – Acronyms and Abbreviations

AACGM	Attitude Adjusted Corrected Geomagnetic
ACE	Advanced Composition Explorer
AFCCC	Air Force Combat Climatology Center
AFOSR	Air Force Office of Scientific Research
AFRL	Air Force Research Laboratory
AFSCN	Air Force Satellite Control Network
AFSPACECOM	Air Force Space Command
AFSWC	Air Force Space Weather Center
AFWA	Air Force Weather Agency
AFWIN	Air Force Weather Information Network
AF/XOW	Air Force Director of Weather
APL	Applied Physics Laboratory of Johns Hopkins University
ASCII	American Standard Code for Information Interchange
ASPAM	Atmospheric Slant Path Analysis Model
AVHRR	Advanced Very High Resolution Radiometer
AVN	Aviation Model
AVO	Alaska Volcano Observatory
BATS	Biosphere-Atmosphere Transfer Scheme
CLASS	Canadian Land Surface Scheme
CME	Coronal Mass Ejections
COE	Common Operating Environment
DII	Defense Information Infrastructure
DMSP	Defense Meteorological Satellite Program
Dst	Disturbance, storm
ECMWF	European Center for Medium-Range Weather Forecasts
EIT	Extreme Ultraviolet Imaging Telescope
EVA	Extravehicular Activities
FAC	Field Aligned Currents
FNMOC	Fleet Numerical Meteorology and Oceanography Center
FSL	Forecast Systems Laboratory
FTP	File Transfer Protocol
GDS	Grid Description Section
GI	Geophysical Institute
GIC	Ground Induced Currents
GIF	Graphic Interchange Format
GIT	Georgia Institute of Technology
GMT	Generic Mapping Tools
GOLD	Geophysical On-Line Data
GOES	Geostationary Operational Environment Satellite
GRIB	Gridded Binary
GSE	Geocentric Solar-Ecliptic
GSFC	Goddard Space Flight Center

GUI	Graphical User Interface
HLBL	High Latitude Boundary Layer
IDL	Interactive Data Language
IMF	Interplanetary Magnetic Field
ISS	International Space Station
JHU	Johns Hopkins University
JHU/APL	Johns Hopkins University/Applied Physics Laboratory
Kp	Planetary Index of Geomagnetic Activity
LAN	Local Area Network
LAPS	Local Analysis and Prediction System
LASCO	Large Angle Spectroscope Coronagraph
LEO	Low-attitude Earth Orbit
LSM	Land Surface Model
MATCH	Model of Atmospheric Transport and Chemistry
MeV	Million Electron Volts
MM5	Fifth Generation Mesoscale Model
NASA	National Aeronautics and Space Administration
NCAR	National Center for Atmospheric Research
NCEP	National Centers for Environmental Prediction
netCDF	Network Common Data Form
NGDC	National Geophysical Data Center
NGM	Nested Grid Forecast Model
NOAA	National Oceanic and Atmospheric Administration
NOGAPS	Navy Operational Global Atmospheric Prediction System
NRL	Naval Research Laboratory
NWP	Numerical Weather Prediction
NWS	National Weather Service
OWS	Operational Weather Squadron
PACE	Polar Anglo-American Conjugate Experiment
PBL	Planetary Boundary Layer
PCA	Polar Cap Absorption
PDS	Product Definition Section
PFRR	Poker Flat Research Range
PNG	Portable Network Graphics
PUFF	From Puff the Magic (ash spewing) Dragon
RBE	Radiation Belt Environment
SAA	South Atlantic Anomaly
SABER	Sounding of the Atmosphere using Broadband Emission Radiometry
SD	Space Department of the Applied Physics Laboratory
SDP	Software Development Plan
SEC	Space Environment Center
SEE	Solar EUV Experiment
SEON	Solar Electro-optical Observing Network
SEP	Solar Energetic Particles
SFOC	Space flight Operations Center

SOHO	Solar and Heliospheric Observatory
SPE	Solar Particle Event
STP	Solar Terrestrial Physics
SWOC	Space Weather Operations Center (Offutt)
SWXS	Space Weather Squadron
SXI	Soft X-ray Imager
Tcl	Tool Command Language
Tk	Toolkit
Tix	Tk Interface Extension
UAF	University of Alaska, Fairbanks
UCAR	University Corporation for Atmospheric Research
UCB	University of Colorado, Boulder
UPOS	University Partnering for Operational Support
UTC	Coordinated Universal Time
WDC	World Data Center
WF	Weather Flight
WMO	World Meteorological Organization
XDR	External Data Representation