PUFF – Volcanic Ash Dispersion Modeling
PUFF-AFWA Version 3.00
Detailed Design Document

Document Number UPOS-B33-03

D. S. Tillman
R. R. Crane

30 Nov 2001

Table of Contents

List of Figures and Tables

1 Introduction

1.1 Purpose

This document is the design document for the PUFF volcanic ash dispersion modeling programs, which predict the geographical distribution of volcanic ash from an eruption versus time. It provides a detailed description of the PUFF suite of programs, the operation of the program package, the relevant technical information, and the capabilities and limitations of PUFF.

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 Design Overview

The "puff" component of the PUFF program suite uses U and V wind velocities versus geopotential height to produce a series of ash files in network common data form (netCDF) format that describe the distribution of the ash at different points in time. U and V wind velocities can be thought of as winds in the easterly and northerly direction respectively on the lat/lon grids. These key inputs must report the U and V wind velocities in meters/second versus pressure levels from 1000 mb (sea level) to about 10 mb.

The “afwa2puff” component of the PUFF program suite is responsible for extracting wind speed data from gridded binary (GRIB) files and writing them to U and V wind speed netCDF files for PUFF’s use.

A single GRIB file represents the prediction of a number of weather characteristics for a given point in time. The data are presented on some type of grid, either lat/lon or XY where each point is related to an actual ground location via a map projection.

Since PUFF needs U and V wind data reported versus geopotential height, afwa2puff must first get U and V wind data versus pressure level, and geopotential height (from the GRIB file also) via pressure level. The U and V wind data are interpolated onto an evenly spaced set of geopotential heights. The final wind files produced for PUFF’s use are U wind data vs. geopotential height, and V wind data vs. geopotential height.

Within a GRIB file, data may be reported for a particular parameter on altitude, pressure level, or sigma level. For PUFF’s case, the data are reported on pressure level.

PUFF requires wind data for a range of times covering the period from a volcanic eruption to the end of the user-requested simulation time. Consequently, the extracted contents of multiple GRIB files are combined into a single netCDF file for U wind data and a single one for V wind data.

1.5 Components

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

Table 1-1 PUFF Suite Executables

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.

1.6 Document Organization

Section 1 describes the scope of the PUFF application system.

Section 2 lists applicable references.

Section 3 provides the detailed design of the afwa2puff program.

Section 4 provides the program flow of the afwa2puff program.

Section 5 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. "AFWA2PUFF - Wind Model Data Conversion Program Functional Requirements Document," Document Number UPOS-B33-03, 13 March 2000
4. “PUFF Input/Output Data Requirements,” Document Number UPOS-B33-02, 12 January 2000
5. “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
6. "wgrib: Portable GRIB Decoder," Wesley Ebisuzaki, http://wesley.wwb.noaa.gov/wgrib.html
7. “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

3 Afwa2puff Design Overview

Identify major structural elements

Afwa2puff comprises a single structural element. It functions as a pre-processor for the PUFF model, converting source wind data in GRIB format into a PUFF-specific configuration in netCDF form (Figure 3-1).

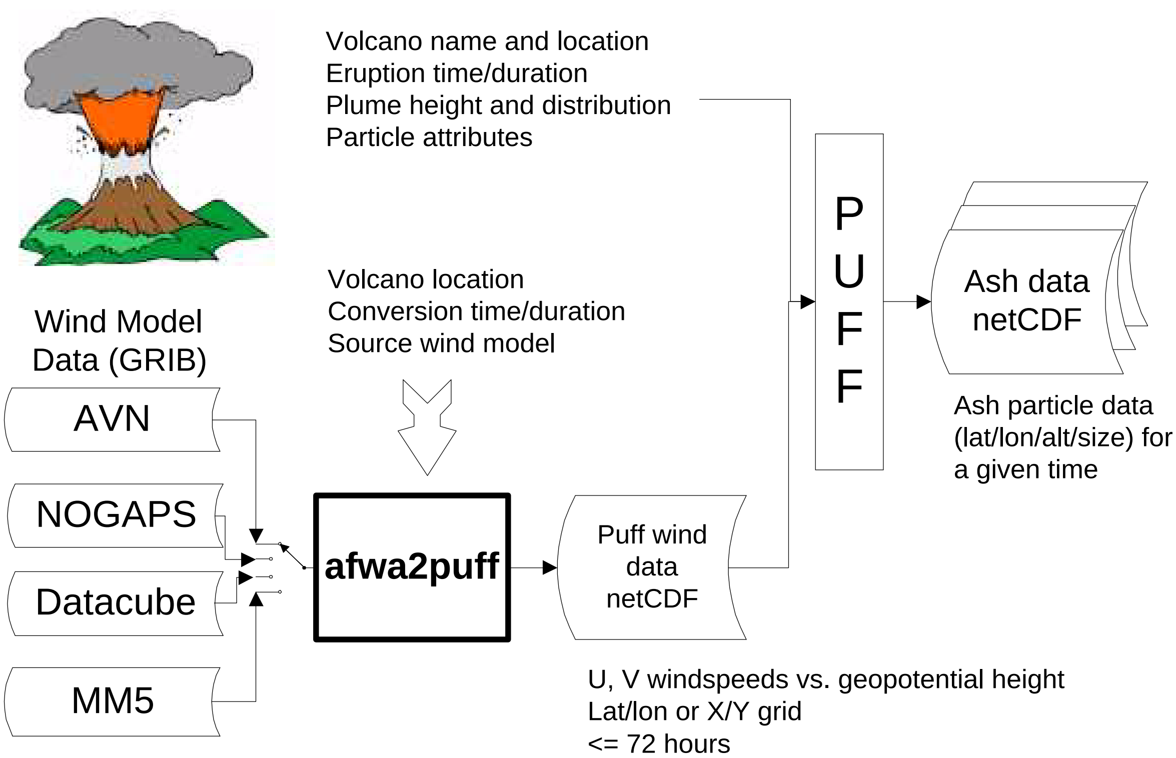

Figure 3-1 PUFF Dataflow Diagram

Identify data and control flow between elements

Another component of the PUFF suite interacts with afwa2puff to effect the wind conversion. The "puffgui" component, a predominantly Tcl-Tk application, provides the user interface through which a user selects the source wind model and identifies the desired time and areal coverage for the wind data. Puffgui captures these selections in a text file, and invokes afwa2puff to perform the conversion. Upon successful completion, puffgui updates its list of available PUFF input wind files. The afwa2puff program has completed its role at that point. This operation is described in Reference [1].

The afwa2puff program can be invoked via the command line. As in the case where it is invoked by puffgui, a text file must be populated with the requisite input parameters, per Reference [2].

List inputs and outputs from each element

Afwa2puff's inputs and outputs are completely described in Reference [2].

4 Afwa2puff Program Flow

The overall program flow is shown in Figure 4-1. These charts depict the high level steps being performed (within each box) and reference function and method calls via a callout if used in the execution of that step. Methods are denoted by the class::method syntax and functions are denoted by the function name. If the function or method are defined in a different file, the filename is listed.

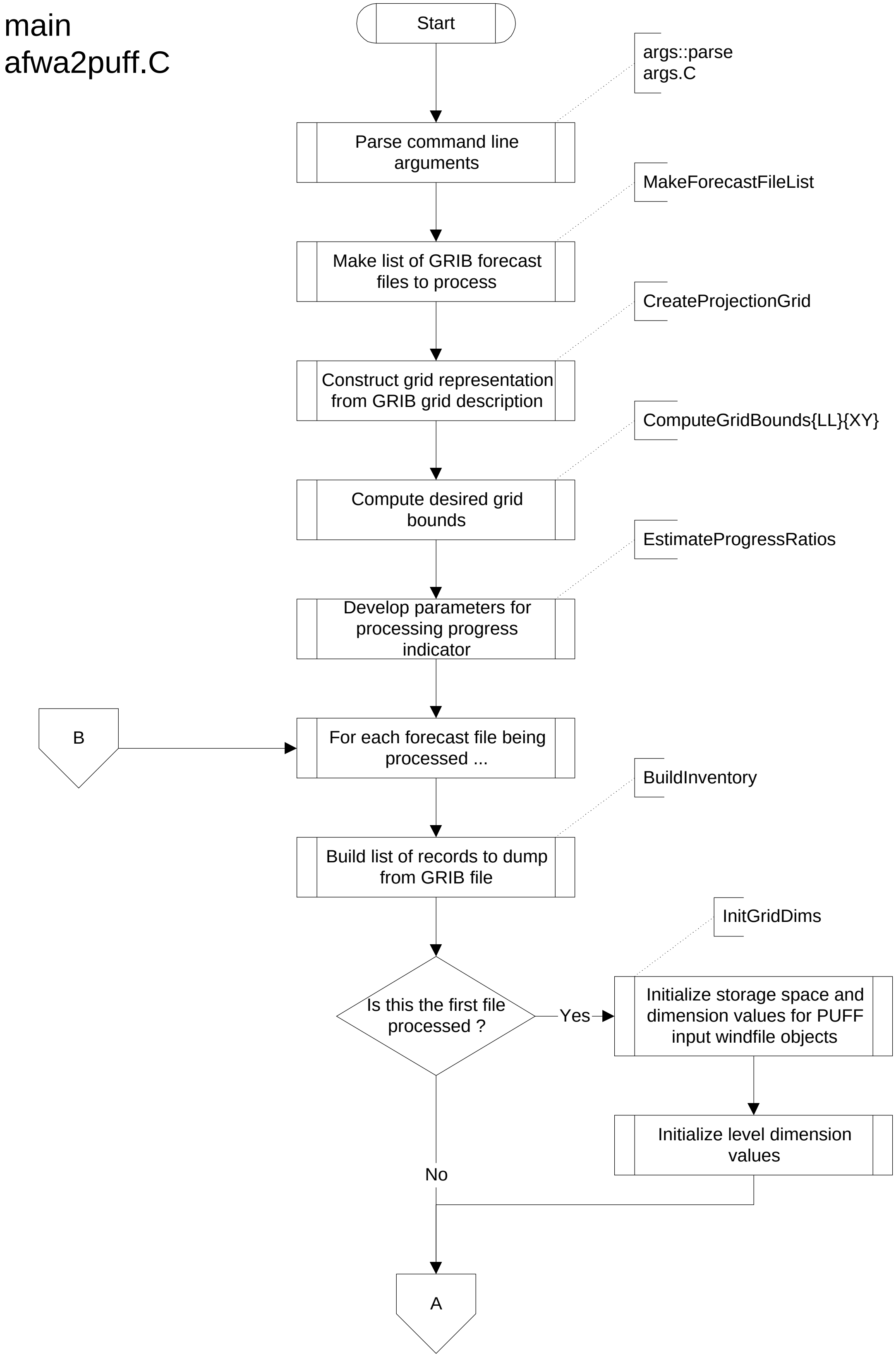

Figure 4-1 Flowchart for afwa2puff main

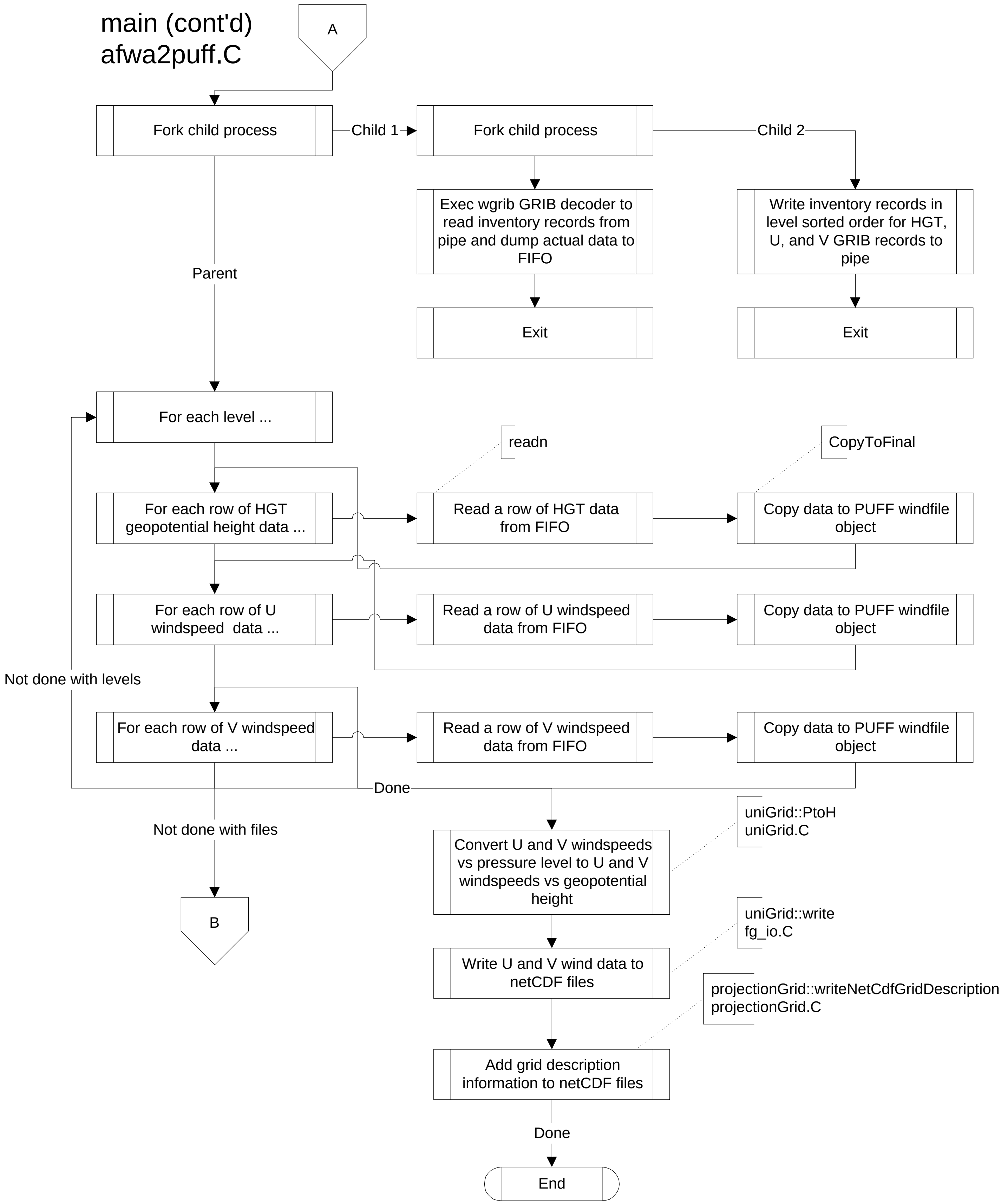

Figure 4-1 Flowchart for afwa2puff main (cont'd)

The majority of the functions within the program can be adequately described via the function headers. For selected cases where this is not true, additional information is provided in this document in subsequent sections. For one case in particular, the logic is sufficiently complicated to warrant a flow chart. This case is the determination of the list of source filenames for the wind data to be converted. The flow chart for this function is given in Figure 4-2.

The MakeForecastFileList function is responsible for creating an array of GRIB filenames in forecast time order that will be used as the source of the wind data. The complexity of the function arises from its general approach and the myriad conditions it needs to handle. The specific requirements are described in Reference [3]. To summarize, the function must:

1. Determine which, if any, base cycles exist for the selected wind model. A base cycle is the time and date of the first data in a forecast run. For some wind models there are two model runs a day; for others there are four; and for others there can be an arbitrary number and they don't have to occur on specific hours.
2. Find the latest base cycle that has forecast coverage for the desired time. That is, the first data in the forecast must be at or earlier than the eruption and the latest data in the cycle must cover at least to the point of volcano eruption time plus the minimum number of user specified hours.
3. For each forecast file in the base cycle, check for presence, readability, and applicability. The program will ignore unusable files and continue to seek usable files until all in the base cycle are exhausted.
4. Create a final list containing only usable files in time order.

Subtle Details the Function Handles

1. MM5 wind data is produced on a variety of grids. The primary grid is termed the "a" grid, is on 36 x 36 km spacing, and will always have forecast data beginning with the base cycle time. However, nested grids (e.g., "b", "c", and "d") for a given model run typically do not have forecast data beginning with the base cycle time. Their first data is usually at least six hours after the base cycle time. So if a user has selected one of the subnests, it's not enough to ensure that the base cycle time occurs at or before an eruption. The actual start of data must be examined. So for a subnest, the program first looks for the base cycle by searching for the "a" nest. Only after identifying the valid base cycles does it look at the start time of the subnest.
2. Weather models are CPU-intensive programs that require many hours to complete processing. They may require 3 or more hours to complete production of all the forecast files in a set, but they make files available as they are ready. Since the filenames generally don't contain date information (only time), they are not unique on a day to day basis. Consequently, what appears to be a full set of forecast files for a run (based on inspection of the filenames) could in fact be a conglomeration of new

files and day-old files. So the program must inspect the internal date information in the file and make sure it matches that of the base cycle file.

3. The interval in hours covered by successive forecast files can vary.
4. Occasionally forecast files can be produced with permissions that preclude the program from accessing them. The operator must be notified but the program shouldn't stop attempting to build the file collection until the user's desired time range is satisfied or the available files for that base cycle are exhausted.

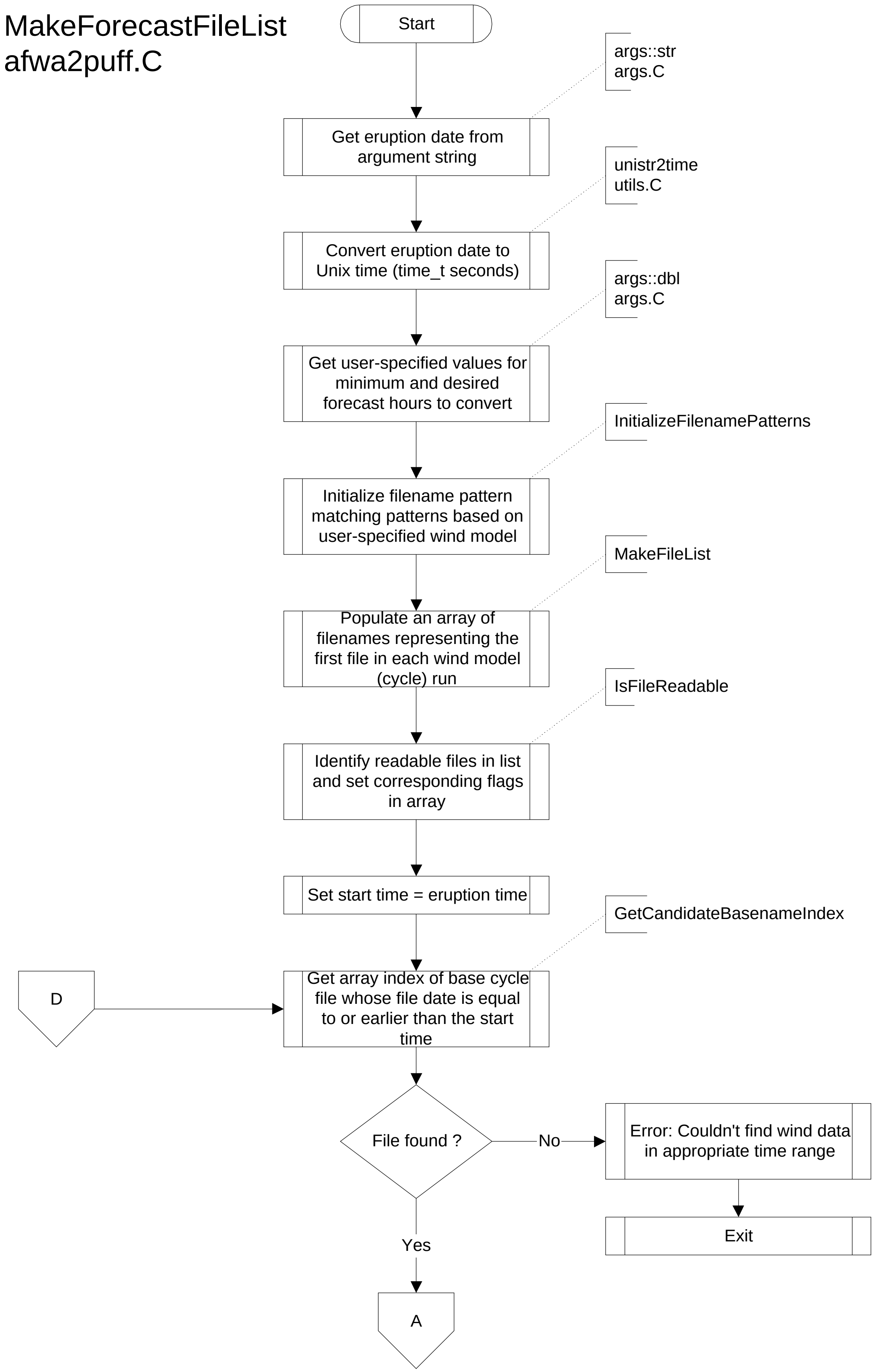

Figure 4-2 Flowchart for MakeForecastFileList

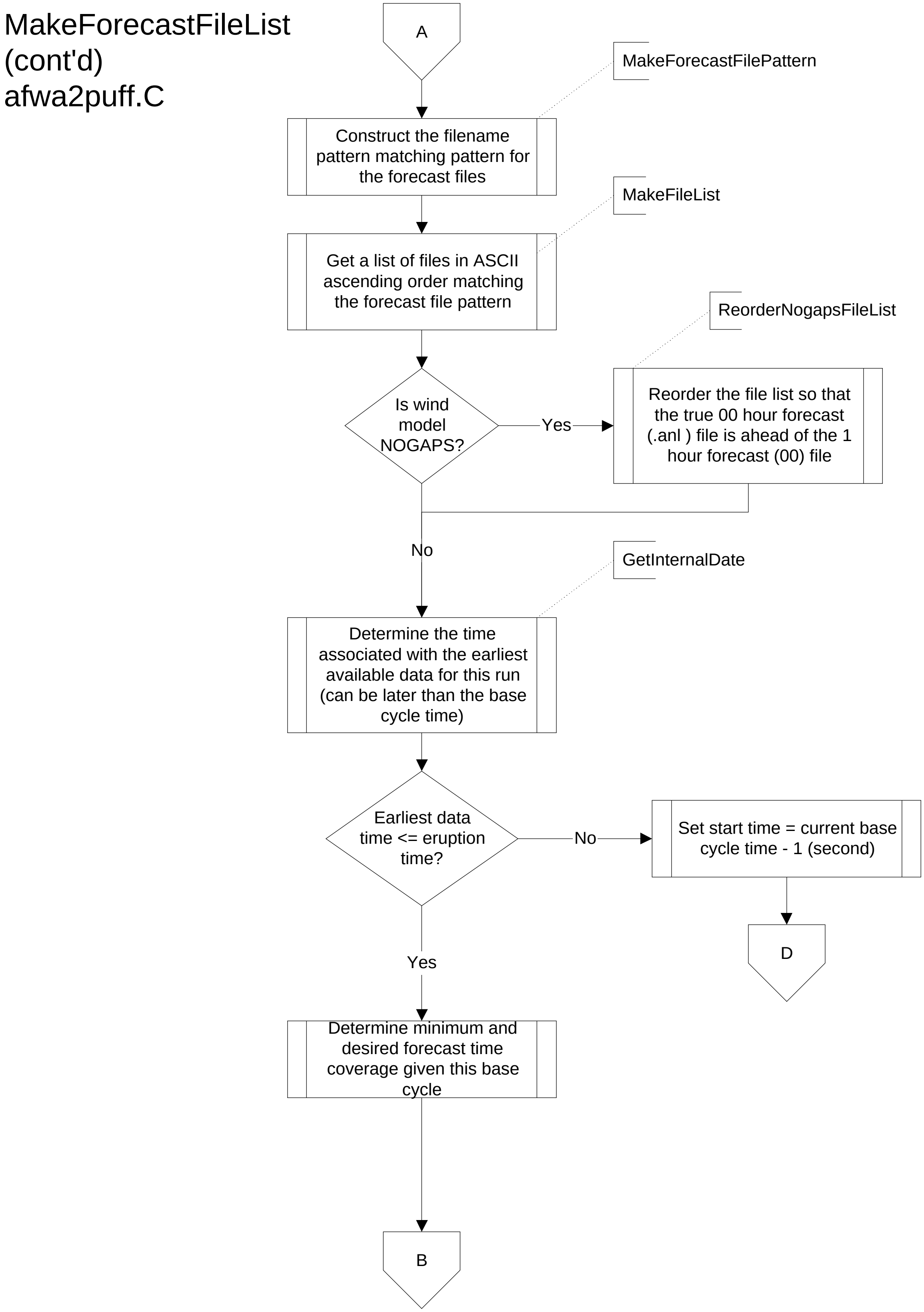

Figure 4-2 Flowchart for MakeForecastFileList (cont'd)

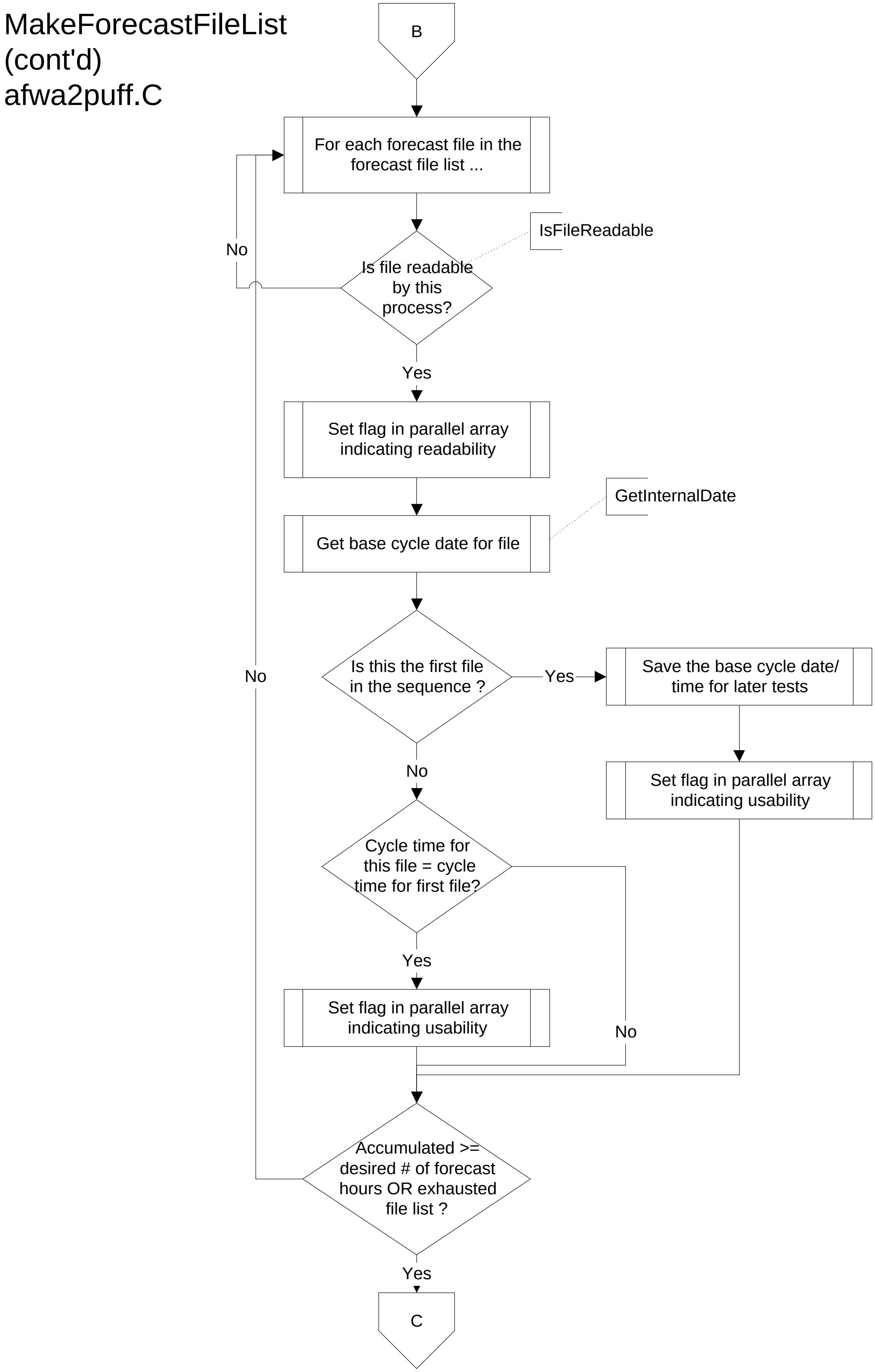

Figure 4-2 Flowchart for MakeForecastFileList (cont'd)

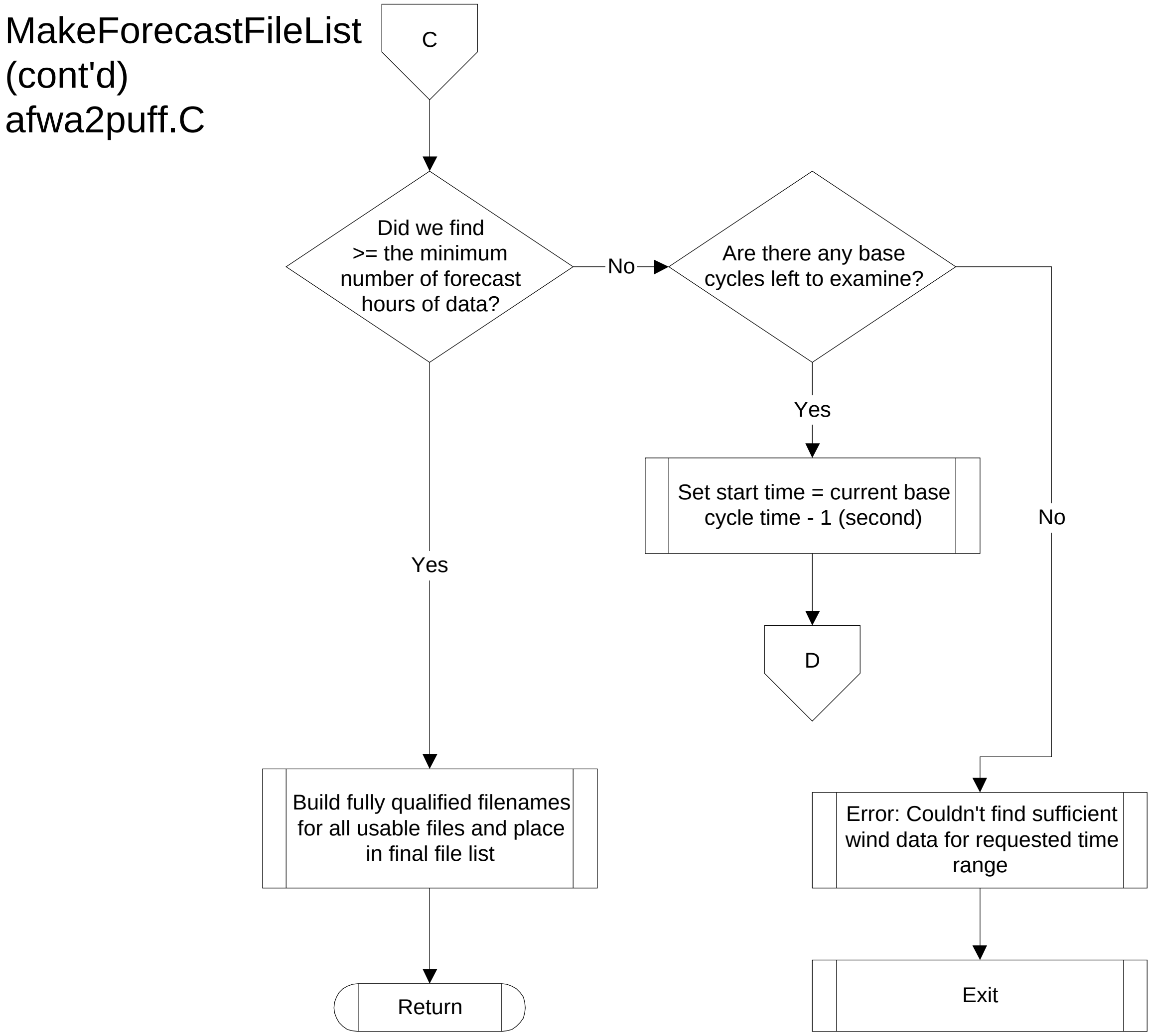

Figure 4-2 Flowchart for MakeForecastFileList (cont'd)

Data are stored in the netCDF file in the following manner:

1. The main variable is the wind speed, and it is stored in a vector. The length of the vector is the product of the number of forecast times, levels, latitude points (or Y grid points), and longitude points (or X grid points). The forecast times, levels, and X and Y parameters form the dimensions.
2. For each dimension, values for the dimension are stored as well. For example, the forecast time vector might look like (0, 3, 6, 9).
3. Additional ancillary information is added to the file to capture information about the original grid and the areal coverage of the wind data. These parameters are used by the GUI wrapper functions to make sure that the input wind files selected by an operator actually cover the volcano for the time period of interest. The grid parameters are also used in the execution of the PUFF model to manage conversions between X,Y grid points and lat/lon points that can be displayed on a map.

(i) <u>Implementation of Modifiable GRIB Filename Descriptions</u>

For each of the possible GRIB model sources, there is a skeletal filename description in the puff.args file. The convention followed is to use a question mark (?) to represent all characters in the GRIB filename that are irrelevant to the conversion and using special single-character codes to indicate the location in the names of the cycle hours and minutes and forecast hours and minutes. The codes are as follows (case matters):

c	cycle hours
d	cycle minutes
f	forecast hours
g	forecast minutes
t	theater
n	nest

Cycle hours and forecast hours are required in all cases. Theater and nest entries are required for the MM5 files. Cycle minutes and forecast minutes are optional (from afwa2puff's perspective). If the filename uses three digits for forecast hours, use three 'f' characters in the skeletal filename. The parsing algorithm expects minutes to follow hours, but not necessarily directly. Other than that, there is no constraint on the location of the key values. Following are some current examples:

NOGAPS:

Base filename: nogaps.T00Z.000
Skeletal filename: ??????.?cc?.fff

Note that in this case the skeletal filename contains periods at the same positions as those in the real filename. This is not required; they could be replaced with question marks. The presence of the periods provides more protection against inadvertent access of unrelated files in the directory.

AVN:

Base filename: gblav.T00Z.PGrbF00
Skeletal filename: ?????.?cc?.?????ff

MM5:

Base filename: us057g1010t03a060000000
Skeletal filename: ???????????ttnccddfffgg

5 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