1/69

SOFTWARE RELEASE NOTICE

01. SRN Number: PA-SRN-150	
02. Project Title: TSPA & I	Project No. 20-5708-761
03. SRN Title: ASHPLUME, Version 1.0	
04. Originator/Requestor: DOE	Date: 06/19/97

05. Summary of Actions

- ☐ Release of new software
- ■ Release of modified software:
  - ☐ Enhancements made
  - ■ Corrections made
- ☐ Change of access software
- ☑ Software Retirement GW 2/12/2003

06. Persons Authorized Access

Name	RO/RW	A/C/D
Keith McConnell	RW	
Mark Jarzemba	RW	
Chuck Connor	RO	
Britt Hill	RO	

07. Element Manager Approval: RG Boca Date: ~~06/19/97~~ 6/24/97 RGB

08. Remarks:

Code has been requested by the DOE.

CNWRA Form TOP-6 (06/95)

ASHPLUME Version 1.0 can be accessed at mammoth:/LAN/RCS/ashplume1.0

2/65

SOFTWARE SUMMARY FORM

01. Summary Date: 06/19/97	02. Summary prepared by (Name and phone) Mark Jarzemba, (210) 522-6828	03. Summary Action: New
04. Software Date: 06/19/97	05. Short Title: ASHPLUME—Code for Ash Dispersal	
06. Software Title: ASHPLUME, Version 1.0		07. Internal Software ID: None

08. Software Type: □ Automated Data System ■ Computer Program □ Subroutine/Module	09. Processing Mode: ■ Interactive □ Batch □ Combination	10. APPLICATION AREA a. General: ■ Scientific/Engineering □ Auxiliary Analyses ■ Total System PA □ Subsystem PA □ Other b. Specific:
11. Submitting Organization and Address: CNWRA 6220 Culebra Road San Antonio, Texas 78238		12. Technical Contact(s) and Phone: Mark Jarzemba, (210) 522-6828

13. Narrative:

ASHPLUME considers physical factors such as eruption energetics, eruption duration, wind velocity, and particle properties to compute the activity areal density as a function of spatial location. Various components of the model are based on empirical relationships and data from observed and monitored cinder cone eruptions.

14. Computer Platform SUN	15. Computer Operating System: Mac, SUN OS, PC	16. Programming Language(s): FORTRAN 77	17. Number of Source Program Statements: 2009
18. Computer Memory Requirements: 1 MB	19. Tape Drives: N/A	20. Disk/Drum Units: N/A	21. Graphics: N/A

22. Other Operational Requirements

NONE

23. Software Availability: ■ Available □ Limited □ In-House ONLY	24. Documentation Availability: ■ Available □ Inadequate □ In-House ONLY

Software Custodian: [signature] Date: 6/23/97

CNWRA Form TOP-4-1

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
QUALITY ASSURANCE
SURVEILLANCE REPORT

PROJECT NO.: 20.01402.159	REPORT NO.: 2000-13	PAGE 1 OF 2

SURVEILLANCE SCOPE: Review of CNWRA Developed Scientific and Engineering Software to determine whether the documentation present in the CNWRA Software Working Records Folders is adequate.

REFERENCE DOCUMENTS: Technical Operating Procedure-018, Development and Control of Scientific and Engineering (S&E) Software; QAP-004, Surveillance Control; Nonconformance Report 2000-03.

STARTING DATE: 3/7/2000 ENDING DATE: 6/9/2000

QA REPRESENTATIVE: B. Mabrito

PERSONS CONDUCTING TEST/EXAM/ACTIVITY: Various CNWRA staff working on Developed S&E software.

SATISFACTORY FINDINGS: During the course of this surveillance, CNWRA Developed S&E software and documentation was checked and contact made with CNWRA staff who worked with the software. In each case, the particular S&E software folder was reviewed for completeness and where no Design Verification Report (DVR) was located, the objective evidence in the folder was compared to the DVR form questions and discussions were held with cognizant CNWRA staff. The list of Developed S&E software reviewed is included in Attachment A.

In each case, key elements of the DVR were compared against that which was included in each software folder in the QA working records. Also, the previous version of the software code documentation was checked to ensure that the earlier DVR had been properly completed. The later version of the software documentation showed the specific changes made through the Software Change Reports. Based on this review, it is clear that although in a few cases no DVR was accomplished, product quality did not suffer. The minor enhancements and "bug" fixes made to TPA Version 3.2.3 and 3DStress Version 1.3.1 and 1.3.2 software were clearly identified and controlled so that the CNWRA product being delivered met the client's requirements.

UNSATISFACTORY FINDINGS: None.

NONCONFORMANCE REPORT NO.: None.

ATTACHMENTS: Attachment A.

RECOMMENDATIONS/ACTIONS: N/A.

APPROVED: [signature]
CENTER DIRECTOR OF QUALITY ASSURANCE

DATE: 6/12/2000

DISTRIBUTION:
ORIGINAL - CENTER QA DIRECTOR QA Records
ORIGINATOR
PRINCIPAL INVESTIGATORS OF EACH CODE
ELEMENT MANAGERS
B. Sagar, H. Garcia

4/69

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

March 12, 1999

Center for Nuclear Waste Regulatory Analyses
ATTN: Dr. Wesley C. Patrick
6220 Culebra Road
Building 189, Room A233
San Antonio, Texas 78238-5166

SUBJECT: QUALITY ASSURANCE DOCUMENTATION OF ASHPLUME

Dear Dr. Patrick:

Sandia National Laboratories (Sandia) staff has requested that U.S. Nuclear Regulatory Commission staff provide the Quality Assurance documentation for the ASHPLUME code that was developed by the Center for Nuclear Waste Regulatory Analyses. Specifically, they are interested in any documentation of Requirements, Test, Design, and Validation. In discussions with Mr. Ralston Barnard of Sandia, we have been informed that this request has been discussed with Mr. Bruce Mabrito of your staff.

This letter is to authorize you to provide Sandia with any existing documentation, either in electronic or hard copy, of the Quality Assurance documentation of the ASHPLUME code. Cost associated with this effort should be charged to Igneous Activity (1402-640). You are not authorized to perform additional services, such as providing matching runs with a U.S. Department of Energy data set. Your contact at Sandia is Mario Chavez at Sandia National Laboratories, Dept. 6811/JHA, MS-0776, P. O. Box 5800, Albuquerque, New Mexico, 87185-0776. He can be reached by telephone at either (505) 844-9073 or (505) 299-1282, and his e-mail address is michavez@nwer.sandia.gov.

If there are any questions regarding this letter, please contact John Trapp at (301) 415-8063, or e-mail at jst@nrc.gov.

Sincerely,

Keith I. McConnell for

Michael J. Bell, Chief
Performance Assessment and High-Level Waste
Integration Branch
Division of Waste Management
Office of Nuclear Material Safety
and Safeguards

cc: B. Mabrito, CNWRA
R. Barnard, Sandia
M. Chavez, Sandia
J. Linehan
B. Meehan

cc: [redacted] BH
LM CC
GW BS

5/69

Center for Nuclear Waste Regulatory Analyses

6220 CULEBRA ROAD • P.O. DRAWER 28510 • SAN ANTONIO, TEXAS, U.S.A. 78228-0510
(210) 522-5160 • FAX (210) 522-5155

March 19, 1999

Sandia National Laboratories
ATTN: Mr. Mario Chavez
Department 6811/JHA, MS-0776
P.O. Box 5800
Albuquerque, New Mexico 87185-0776

Subject: Quality Assurance Documentation of the ASHPLUME Software

Dear Mr. Chavez:

The purpose of this letter is to convey to you the Quality Assurance documentation for the ASHPLUME code that was developed by the Center for Nuclear Waste Regulatory Analyses.

Enclosed is a copy of the following items:

- ASHPLUME Consequence Module Code Software Requirements Description
- A copy of the ASHPLUME Design Verification Report
- Copies of the ASHPLUME Software Summary Form and Software Release Notice
- Copies of additional ASHPLUME Software Tests and NRC/DOE correspondence
- A copy of the ASHPLUME Software and related Technical Description and User's Guide.

Please call me if you have any questions regarding this transfer of material. I can be reached at (210) 522-5149.

Very truly yours,

Bruce Mabrito
Director of Quality Assurance

cc: W. Patrick, CNWRA, w/o attachments
B. Meehan, NRC Contracts, w/o attachments
M. Bell, NRC DWM, w/o attachments
D. DeMarco, NRC PMDA, w/o attachments
B. Stiltenpole, NRC PMDA, w/o attachments
CNWRA Directors/Element Managers, w/o attachments
S. Boyanowski, SwRI Contracts, w/o attachments

h:\mabrito\doechavez.ltr

6/69

Subject Code 672
Project No. 20-1402-159

Sandia National Laboratories
Operated for the U.S. Department of Energy by
Sandia Corporation

P.O. Box 5800
Albuquerque, NM 87185-0778

Phone: (505)284-4605
Fax: (505)284-4002
Internet: rwbarna@sandia.gov

WBS: 1.2.1.12
QA: NA

February 25, 1999

Copies to
W Patrick
Director
EMS
Ashplume Software
Folder

Mr. Michael J. Bell
U.S. Nuclear Regulatory Commission
11545 Rockville Pike
Two White Flint
Mail Stop 7 F3, Room 7 F2
Rockville, Maryland 20852-2738

Dear Mr. Bell:

Subject: ASHPLUME Quality Assurance Code

As you may be aware, Sandia National Laboratories (SNL) is in the process of qualifying the ASHPLUME code for use on the Yucca Mountain Project (YMP). Since any major software used on the YMP has to be appropriately qualified, there are some procedural requirements we have to follow. One is that testing be performed to demonstrate adequacy with the code requirements. To date, SNL has been able to replicate one of the tests provided in CNWRA-97-004 Rev. 1. However, SNL would like to obtain the entire ASHPLUME Software Quality Assurance (SQA) configuration to do a complete job of testing. Our SQA Staff has contacted the Southwest Research Institute (SWRI) about obtaining the required documentation and SWRI is amenable to that request provided they are given instructions from the NRC to do so. Therefore, SNL would like to request that SWRI provide SNL with the needed ASHPLUME documentation, specifically the Requirements, Test, Design, and Validation documentation. This documentation will make Sandia's qualification effort more effective, as well as save resources for other priority tasks.

Mario Chavez from our SQA staff will serve as the point of contact for the transfer of any SQA documentation. Mr. Chavez can be reached at Sandia National Laboratories, Dept. 6811/JHA, MS-0776, P.O. Box 5800, Albuquerque, NM 87185-0776, telephone 505-844-9073 or 505-299-1282, e-mail mjchavez@nwer.sandia.gov.

Sincerely,

Ralston W. Barnard

File: Ashplume 7/69
Software
QA
Folder

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

RECEIVED CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

July 15, 1997

013430 JUL 21 97

SUBJECT CODE 707.2
PROJECT NO. 20-5708-762

Dr. Robert G. Baca
Performance Assessment Element Manager
Center for Nuclear Waste Regulatory Analyses
6220 Culebra Road
San Antonio, Texas 78228-0510

SUBJECT: ACCEPTANCE OF TECHNICAL DESCRIPTION & USER'S GUIDE FOR ASHPLUME MODULE - INTERMEDIATE MILESTONE: 5708-762-720

Dear Dr. Baca:

We have evaluated and accept the CNWRA submission of the revised "Technical Description and User's Guide for ASHPLUME Module" that was submitted on June 24, 1997. We note that changes to the ASHPLUME module were made to reflect recent changes to the mathematical model implemented in the code, and that the driving force of this activity was the DOE request to obtain a copy of the ASHPLUME code.

If you have any questions, please contact me at (301) 415-7289.

Sincerely,

Keith I. McConnell, Element Manager
Total System Performance Assessment and Integration KTI
Division of Waste Management
Office of Nuclear Material Safety and Safeguards

cc: J. Linehan, PMDA
B. Meehan, CMB1/ADM

cc. W. Patrick
Directors
EMs
M. Jarzemba
P. La Plante

8/69

Mr. Mabrito; 24 J[illegible] 1997

Herewith notification that a S&E Software code has been placed under Version Control in accordance with TOP-018, para.5.7.

Details

Program Name: ASHPLUME V 1.0
Category: CNWRA developed
Developer: M. Jarzemba
Date Entered: 24 June 97
Control Method: SCCS
Location: mammoth:/lan/rcs/ashplume1.0

Chris Wilkinson
Pico Consulting
ext-5238

```
# Original source code for ashplume 1.0 as controlled by SCCS.
# June 24, 1997
drwxr-xr-x   2 root     other        512 Jun 24 13:32 SCCS
-r--r--r--   1 root     other      92182 Jun 24 13:31 ashplume.f
-r--r--r--   1 root     other        809 Jun 24 13:31 ashplume.in
-r--r--r--   1 root     other       5024 Jun 24 13:31 ashplume.out
-r--r--r--   1 root     other        604 Jun 24 13:32 finder.dat
```

```
# History files for ashplume1.0 under SCCS control
# June 24, 1997
#
-r--r--r--   1 root     other     127160 Jun 24 13:31 s.ashplume.f
-r--r--r--   1 root     other       1266 Jun 24 13:31 s.ashplume.in
-r--r--r--   1 root     other       7074 Jun 24 13:31 s.ashplume.out
-r--r--r--   1 root     other        986 Jun 24 13:32 s.finder.dat
```

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

11/69

March 12, 1997

Dr. Robert G. Baca
Performance Assessment Element Manager
Center for Nuclear Waste Regulatory Analyses
6220 Culebra Road
San Antonio, Texas 78228-0510

Dear Dr. Baca:

SUBJECT: ACCEPTANCE OF INTERMEDIATE MILESTONE: 5708-762-720, ASHPLUME VERSION 1.0 TECHNICAL DESCRIPTION AND USER'S GUIDE - LETTER REPORT

We have reviewed and accept the CNWRA Letter Report documenting the completion of the ASHPLUME module. We acknowledge that this code is currently in a stand-alone form, but will be incorporated into the TPA 3.0 code by the March 17, 1997, deliverable date for the system code. As part of that incorporation, the input, parameter sampling, and output of the stand-alone version will be appropriately modified such that the ASHPLUME module can be exercised from the executive level of the TPA code. This new interface will also be documented in the User's Guide for the TPA code.

If you have any questions, please contact me at (301) 415-7289.

Sincerely,

Keith I. McConnell

Keith I. McConnell, Element Manager
Total System Performance Assessment and Integration KTI
Division of Waste Management
Office of Nuclear Materials Safety and Safeguards

cc: B. Meehan, CAB1/ADM
J. Linehan, PMDA

cc: W. Patrick
Directors
Element Managers
M. Jarzemba

12/69

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE: ASHPLUME V. 1.0

June 23, 1997

ASHPLUME VERSION 1.0

1. Scientific Notebook Documentation Development: CNWRA Scientific Notebook Number 164 was verified and it documented the software development of ASHPLUME Version 1.0.

2. Programming Language: ANSI Standard FORTRAN 77 confirmed by the Software Custodian; ASHPLUME software compiled on SUN OS FORTRAN compiler and a MAC OS system.

3. Internal Documentation: On 6/20/97, B. Mabrito reviewed portions of the ASHPLUME Version 1.0 software on the M. Jarzemba CNWRA computer platform. Under the subroutine function file names of "ashcalca," "ashcalcf," "dintegrandf," and "FFintegrand," the comment lines were reviewed and there was clear and numerous internal documentation comments meeting the requirements of TOP-018 Section 5.4.4. A hard copy of some of the comments were made and will become part of this package.

4. Software Labels and Data

a. Header Data and Format: The ASHPLUME Version 1.0 program main module header data and the format were compared against TOP-018 requirements and found acceptable. It should be noted that the ASHPLUME Version 1.0 software developer added the contractually required disclaimer information during the design verification process.

b. NRC Data: The ASHPLUME Version 1.0 header data and the format were compared against TOP-018, Section 5.4.6, third bullet and found acceptable, except that the NRC "user" does not show a specific NRC contact because none has been identified. This work evolved from an earlier CNWRA Research Project and has been "carried" by the Performance Assessment Element.

c. Source Code Header: The ASHPLUME Version 1.0 source code header data provided the minimum requirements of TOP-018, Section 5.4.6, fourth bullet and found acceptable. Software Problem/Change Reports relating to ASHPLUME are included in this package.

5. Unique Run Identification: At the current time the ASHPLUME Version 1.0 software is configured to meet the TOP-018 Section 5.4.5 requirement of having an identifier so that individual executions are uniquely identified. The software developer stated that with the "date/time requirement written into the software" it makes the code "platform specific" and one of the goals of CNWRA scientific and engineering software is "portability." It was decided by the developer to include the capability of entering the date/time printing feature, but the portability issue is important enough to leave out that automatic date/time function date/time identifier in ASHPLUME Version 1.0. It should be made clear that the date and time does clearly show at the top of each hard copy page attached because the ASHPLUME format allows the user to insert such information.

6. Software Analysis and Results

a. Analysis: FOR_STUDY Version 1.2 was utilized as the software analysis tool by R. Janetzke for the developer M. Jarzemba. FOR_STUDY was run in the "default configuration" as specified in the Users Manual. Additionally, M. Jarzemba utilized "Absoft Fx Debugger," a software development tool and utilized feedback to make corrections throughout development of the ASHPLUME Version 1.0 code. A copy of the Absoft FX Debugger general description is attached to this documentation.

b. Analysis Report: Less than ten (10) warning messages were identified by the FOR_STUDY analysis. There were no error messages from the FOR_STUDY exercise. This is documented in Scientific Notebook Number 164 and in an E-mail message from M. Jarzemba to B. Mabrito dated 6/23/97. This is an extremely low number of warning messages compared to other CNWRA codes (over 2,000 warning messages resulted from one FOR_STUDY run on another code). The FOR_STUDY work was coordinated between R. Janetzke and M. Jarzemba.

c. Resolution of Comments: The developer of ASHPLUME reviewed all of the warning messages and addressed all of them by changing the order of the variables in the call. No other action needed to be taken as far as resolution of comments.

Mark A. Jarzemba 6/23/97
Mark Jarzemba, Software Developer **Date**

Bruce Mabrito 6/23/97
Bruce Mabrito, Software Custodian **Date**

Attachments/

original to: Software Folder
cc: CNWRA Software Developer

14/69

To: Bruce Mabrito at CNWRA-OS2
From: Mark Jarzemba
Subject: FOR_STUDY analysis of ASHPLUME
06-23-97 09:12 AM

This purpose of this e-mail is to document statements made to B. Mabrito from M. Jarzemba about a FOR_STUDY analysis that R. Janetzke conducted on the ASHPLUME code.

R. Janezke ran FOR_STUDY on the subject code, and about 8 warning messages were developed from the run of FOR_STUDY. The messages were one of two 'types':

(i) common block called out in subroutines when none of the variables was used.

(ii) ordering of variables in subroutine call statements was not input first then output.

Both types of warnings were eliminated. No errors were found.

Mark Jarzemba

15/69

postscript

JOB 491

Input/output files

MSJ: :ashplume.in

For:	**MSJ**
Creator:	**MPW Shell: LaserWriter 8 8.1.1**
Creation Date:	**1:03 PM Friday, June 20, 1997**

Submit queue:	IF 1 / Ethernet / ETHERTALK
Submitted:	Sun Apr 27 05:03:22 1992
Started:	Sun Apr 27 05:03:23 1992

QMS

QMS 3825 Print System

QMS 3825 Print System

16/69

```
Example 1. User input case w/ beta = 0.02; 6/20/97 11:35 am
0.d0  0.d0                          ! xmin, xmax in km
-1.d0   -50.d0                      ! ymin, ymax in km
1                                   ! numptsx
50                                  ! numptsy
4.80d0  6.86d0                      ! tlogmin, tlogmax- logs of t in sec
9.41    11.55                       ! powlogmin, powlogmax- logs of P in W
-2.0d0  -0.3d0                      ! betalogmin, betalogmax-logs
-2.0d0  -1.0d0  1.0d0               ! dmeanmin, dmeanmed, dmeanmax-logs of d in cm
0.1d0   1.0d0                       ! dsigmamin, dsigmamax
0.8     2.5                         ! ashdenmin, ashdenmax in g/cm3
-2.d0   -1.d0                       ! ashrholow, ashrhohi
0.5d0                               ! fshape
0.001293d0      1.8d-04             ! airden in g/cm3, airvis in g/cm-s
400.d0                              ! c in cm2/s to the 5/2
10.d0                               ! dmax in cm
0.01d0 0.1d0 1.d0                   ! fdmin, fdmean, fdmax al in cm
0.001d0                             ! hmin in km
1.d-10d0                            ! acutoff in g/cm2
0.3d0                               ! rhocut-incorporation ratio
1.d7                                ! Uran- total mass of fuel in g
```

17/69

postscript

JOB 493

Input/Output Files

MSJ: :ashplume.out

For:	**MSJ**
Creator:	**MPW Shell: LaserWriter 8 8.1.1**
Creation Date:	**1:04 PM Friday, June 20, 1997**
Submit queue:	IF 1 / Ethernet / ETHERTALK
Submitted:	Sun Apr 27 05:04:35 1992
Started:	Sun Apr 27 05:04:36 1992

QMS

QMS 3825 Print System

QMS 3825 Print System

18/69

iseed= 10
ASHPLUME version 1.0
Example 1. User input case w/ beta = 0.02; 6/20/97 11:35 am

```
*********************************************************************
*                                                                   *
*                    realization number                    1        *
*                     wind speed (cm/s)            1000.0000        *
*                  wind direction (deg)             -90.0000        *
*           mean particle diameter (cm)                .1000        *
*                          log- std dev                .4000        *
*                       column ht (km)              10.0000        *
*                    event duration (s)          0.3318E+06        *
*                          ash mass (g)          0.1000E+16        *
*                       event power (W)          0.2212E+13        *
*                                 beta                .0200        *
*             vent exit velocity (cm/s)          10000.0000        *
*              particle shape parameter               .5000        *
*                    air density (g/cc)          0.1293E-02        *
*                 air viscosity (g/cm-s)         0.1800E-03        *
*        eddy diff. constant (cm2/s5/2)            400.0000        *
*                      size cutoff (cm)             10.0000        *
*                   incorporation ratio                .3000        *
*        fuel particle minimum log-diam             -2.0000        *
*         fuel particle median log-diam             -1.0000        *
*        fuel particle maximum log-diam                .0000        *
*         total fuel mass available (g)          0.1000E+08        *
*                                                                   *
*********************************************************************
```

x (km)	y (km)	xash (g/cm^2)	xfuel (g/cm^2)
.000	-1.000	0.1904E+05	0.2635E-03
.000	-2.000	0.7632E+04	0.1174E-03
.000	-3.000	0.4915E+04	0.7326E-04
.000	-4.000	0.3601E+04	0.5036E-04
.000	-5.000	0.2694E+04	0.3639E-04
.000	-6.000	0.2178E+04	0.2694E-04
.000	-7.000	0.1798E+04	0.2020E-04
.000	-8.000	0.1502E+04	0.1523E-04
.000	-9.000	0.1269E+04	0.1153E-04
.000	-10.000	0.1083E+04	0.8747E-05
.000	-11.000	0.9277E+03	0.6650E-05
.000	-12.000	0.7962E+03	0.5062E-05
.000	-13.000	0.6857E+03	0.3859E-05
.000	-14.000	0.5910E+03	0.2969E-05
.000	-15.000	0.5093E+03	0.2251E-05
.000	-16.000	0.4388E+03	0.1724E-05
.000	-17.000	0.3778E+03	0.1323E-05
.000	-18.000	0.3248E+03	0.1015E-05
.000	-19.000	0.2785E+03	0.7790E-06
.000	-20.000	0.2356E+03	0.5978E-06
.000	-21.000	0.2029E+03	0.4581E-06
.000	-22.000	0.1723E+03	0.3498E-06
.000	-23.000	0.1458E+03	0.2659E-06
.000	-24.000	0.1229E+03	0.2005E-06
.000	-25.000	0.1032E+03	0.1506E-06
.000	-26.000	0.8647E+02	0.1120E-06
.000	-27.000	0.7231E+02	0.8321E-07
.000	-28.000	0.6035E+02	0.6199E-07
.000	-29.000	0.5036E+02	0.4575E-07
.000	-30.000	0.4206E+02	0.3336E-07
.000	-31.000	0.3520E+02	0.2401E-07

19/69

.000	-32.000	0.2954E+02	0.1592E-07
.000	-33.000	0.2489E+02	0.1117E-07
.000	-34.000	0.2107E+02	0.7797E-08
.000	-35.000	0.1793E+02	0.5418E-08
.000	-36.000	0.1536E+02	0.3744E-08
.000	-37.000	0.1319E+02	0.2599E-08
.000	-38.000	0.1141E+02	0.1799E-08
.000	-39.000	0.9926E+01	0.1247E-08
.000	-40.000	0.8675E+01	0.8665E-09
.000	-41.000	0.7618E+01	0.6034E-09
.000	-42.000	0.6719E+01	0.4212E-09
.000	-43.000	0.5952E+01	0.2946E-09
.000	-44.000	0.5293E+01	0.2064E-09
.000	-45.000	0.4724E+01	0.1447E-09
.000	-46.000	0.4231E+01	0.1015E-09
.000	-47.000	0.3801E+01	0.7211E-10
.000	-48.000	0.3426E+01	0.5072E-10
.000	-49.000	0.3096E+01	0.3553E-10
.000	-50.000	0.2805E+01	0.2491E-10

20/69

postscript

JOB 490

MSJ: :ashplume.f

For:	**MSJ**
Creator:	**MPW Shell: LaserWriter 8 8.1.1**
Creation Date:	**1:01 PM Friday, June 20, 1997**

Submit queue:	IF 1 / Ethernet / ETHERTALK
Submitted:	Sun Apr 27 05:02:00 1992
Started:	Sun Apr 27 05:02:06 1992

QMS

QMS 3825 Print System

QMS 3825 Print System

21/69

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
c                                                                        c
        Program ashplume
c                                                                        c
c                                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
c     code: ashplume (spent fuel incorporation and ash dispersion        c
c           code)                                                        c
c                                                                        c
c     version: 1.0                                                       c
c                                                                        c
c     date: 6/19/97                                                      c
c                                                                        c
c     purpose:                                                           c
c                                                                        c
c     This program calculates the ash and incorporated fuel              c
c     areal density at the x,y points on the earth's surface after       c
c     a volcanic event, assumed to penetrate the proposed Yucca          c
c     Mountain repository exumes spent nuclear fuel                      c
c                                                                        c
c                                                                        c
c     The user is given the several options for what type of             c
c     information that they want to have calculated. The code            c
c     can be run in stochastic mode, or can be run in a user input       c
c     mode where the user inputs all of the required informaton          c
c     for calculating the isopachs. The particle size distributions      c
c     of ash and spent-fuel at the (x,y) points of interest can also     c
c     be generated if so desired.                                        c
c                                                                        c
c     Operation of this code is described in 'ASHPLUME Version 1.0:      c
c     A Code for Contaminated Ash Dispersal and Deposition- Technical    c
c     Descritption and User's Guide'. CNWRA 97-004 Revision 1. June      c
c     1997. Center for Nuclear Waste Regulatory Analyses.                c
c     San Antonio, Tx.                                                   c
c                                                                        c
c  Developer: Mark S. Jarzemba                                           c
c             Research Engineer                                          c
c             Performance Assessment Element                             c
c             Center for Nuclear Waste Regulatory Analyses               c
c             San Antonio, TX  78245                                     c
c             (210) 522-6828                                             c
c                                                                        c
c  User:      Nuclear Regulatory Commission                              c
c             Office of Nuclear Material Safety and Safegaurds           c
c             Division of Waste Management                               c
c                                                                        c
c  Disclaimer:This computer code was prepared as as account of work      c
c             performed by the Center for Nuclear Waste Regulatory       c
c             Analyses (CNWRA) for the Division of Nuclear Waste         c
c             Management of the Nuclear Regulatory Commission (NRC),     c
c             an independent agency of the United States Government.     c
c             The developer of the code, nor any sponsors make any       c
c             warranty, expressed or implied, or assume any legal        c
c             responsibility for the accuracy, completeness, or          c
c             usefulness of any information, apparatus, product or       c
c             process disclosed, or represent that its use would         c
c             not infringe on privately-owned rights.                    c
c                                                                        c
c             In no event unless required by applicable law will         c
c             the sponsors or those who have written or modified         c
c             this code, be liable for damages, including any lost       c
```

22/69

```
c             profits, lost monies, or other special, incidental or     c
c             consequential damages arising out of the use or           c
c             inability to use (including but not limited to loss       c
c             of data or data being rendered inaccurate or losses       c
c             sustained by third parties or a failure of the program    c
c             to operate with other programs) the program, even if      c
c             you have been advised of the possibility of such damages  c
c             or for any claim by another party.                        c
c                                                                       c
c                                                                       c
c         maxd = maximum dimension of isopach points                    c
c         nmy  = maximum number of volcanoes                            c
c                                                                       c
      parameter (maxd=500)
      parameter (nmy=1000)
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      character*60 version,title
      common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
      common /two/ h,werupt0,airden,airvis,c,u
      common /three/ fdmin,fdmax,fdmean,hmin,hmax,xmin,xmax
      common /three1/ dmin,dmax,rhomin,rhomax,rhomean
      common /four/ ymin,ymax,acutoff
      common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
      common /six/ numptsx,numptsy
      common /seven/ Uran
      common /eight/ rhocut
      common /nine/ v,icount
      common /eleven/ ashrholow,ashrhohi
      common /twelve/ power,tdur
      common /thirteen/ numapts
      common /fifteen/ ipchar
      common /sixteen/ title
      dimension v(10000),xash(maxd,maxd),xfuel(maxd,maxd)
      open(unit=10,file='ashplume.out',status='unknown')
      version='ASHPLUME version 1.0'
c                                                                       c
c         Uran    = amount of waste extruded (1 container ~ 10MTU)      c
c         icount  = a counter for the string of random numbers          c
c                   stored in v                                         c
c         iseed   = the random number seed                              c
c         max     = the number of random numbers to get- 8 random nos.  c
c                   per realization                                     c
c         numapts = the number of particle sizes to use in the          c
c                 histogram of particle size at the dose pt.            c
c                                                                       c
      icount=1
      iseed=10
      numapts=30
      write (10,*)'iseed= ',iseed
      write (10,*) version
      max=10000
      pi=dacos(-1.d0)
      call rand(iseed,v,max)
      ichoice=0
c                                                                       c
c         ask the user for their choice of either a number of           c
c         stochastically sampled volcanos or a specific input parameter c
c         set for one volcano                                           c
c                                                                       c
      do while (ichoice.eq.0)

         write (6,31) '**************************************************
```

```
     &**********************'
          write (6,30) '*',' ','*'
           write (6,30) '*','  Do you want stochastic sampling of numerou
     &s volcanoes (ENTER 1)    ','*'
           write (6,30) '*',' ','*'
           write (6,30) '*','                                OR
     &                        ','*'
                write (6,30) '*',' ','*'
           write (6,30) '*','   one volcano for  a  specific input  param
     &eter  set   (ENTER 2)?    ','*'
           write (6,30) '*',' ','*'
           write (6,30) '*','   For the stochastic sampling routine,the c
     &ode will take input      ','*'
           write (6,30) '*','    data from the file -ashplume.in-; else you
     & will be asked to      ','*'
           write (6,30) '*','      input some of the parameters from the
     & keyboard             ','*'
          write (6,30) '*',' ','*'
           write (6,31) '**************************************************
     &***********************'
30         format (a1,a68,a1)
31         format (a70)
          read (5,*) ichoice
           if (ichoice.ne.1.and.ichoice.ne.2) then
             write (6,*)
             write (6,*)'*** TRY AGAIN- must  enter a 1 or a 2 ***'
             write (6,*)
             ichoice=0
           end if
         end do
         if (ichoice.eq.1) then
           write (6,*) 'Number of volcanoes to evaluate? (max=1000)'
          read (5,*) num
         else
           num=1
         end if
         write (6,*)
         write (6,*)'Would you like particulate size information at
     & the dose points?'
        write (6,*)
11         write (6,*)'Enter 1 for yes or 2 for no'
         read (5,*) ipchar
         if (ipchar.ne.1.and.ipchar.ne.2) then
           write (6,*)' you must enter a 1 or 2'
          go to 11
         end if
c                                                                       c
c         time to calculate xash, xfuel in g/cm**2                      c
c                                                                       c
           do k=1,num
c                                                                       c
c         inputdata selects the parameters for the realization          c
c                                                                       c
             call inputdata
             if (k.eq.1) then
               write(10,333) title
333               format(a60)
               write(10,*)
             end if
             if (ichoice.eq.2) then
               call userinput
             end if
             call outheader (k)
```

24/69

```
                go to 20
              end if
              blahold=blah
              y=y+deltay
            end do
20            continue
            if (temp2.eq.0.d0.and.itemp1.eq.1) then
              write(6,*)
              write(10,*)
              go to 10
            end if
            x=x+deltax
            write (6,*)
            write (10,*)
          end do
10          continue
        end do
      close(unit=10)
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      subroutine outheader (ivol)
c                                                                      c
c       This subroutine outputs the header blocks                      c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      parameter (nmx=200, nmy=200)
      parameter (maxd= 200)
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
      common /two/ h,werupt0,airden,airvis,c,u
      common /three/ fdmin,fdmax,fdmean,hmin,hmax,xmin,xmax
      common /three1/ dmin,dmax,rhomin,rhomax,rhomean
      common /four/ ymin,ymax,acutoff
      common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
      common /six/ numptsx,numptsy
      common /seven/ Uran
      common /eight/ rhocut
      common /twelve/ power,tdur
c                                                                      c
c       write the volcanic parameters to the screen                    c
c                                                                      c
      write (6,120) '***********************************
     &*************************************************'
     write (6,130) '*',' ','*'
      write (6,140) '*          ','realization number',ivol,'*'
      write (6,100) '*          ','wind speed (cm/s)',u,'*'
      write (6,100) '*          ','wind direction (deg)',udir,'*'
      write (6,100) '*          ','mean particle diameter (cm)',
     & dmean,'*'
      write (6,100) '*          ','log- std dev',dsigma,'*'
      write (6,100) '*          ','column ht (km)',hmax,'*'
      write (6,110) '*          ','event duration (s)',tdur,'*'
      write (6,110) '*          ','ash mass (g)',q,'*'
      write (6,110) '*          ','event power (W)',power,'*'
      write (6,100) '*          ','beta',beta,'*'
      write (6,100) '*          ','vent exit velocity (cm/s)',
     & werupt0,'*'
      write (6,100) '*          ','particle shape parameter',fshape,
     & '*'
      write (6,110) '*          ','air density (g/cc)',airden,'*'
      write (6,110) '*          ','air viscosity (g/cm-s)',
```

25/69

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
      subroutine ashcalca(ash,ivol)
c                                                                        c
c      This subroutine calculates the ash densities at the dose pts.     c
c                                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      external dinnerf,dinnera
      common /threel/ dmin,dmax,rhomin,rhomax,rhomean
      common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
      common /thirteen/ numapts
c                                                                        c
c      do integration up to rhomax or a max corresponding to the         c
c      limit in the input file depending on which is greater             c
c                                                                        c
      cmax=rhomax
      blah=dlog10(dmax)
      if (cmax.ge.blah) then
        cmax=blah
      end if
      call qrombl(dinnera,rhomin,cmax,ash,x,y,ivol,numapts)
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
      subroutine ashcalcf(fuel)
c                                                                        c
c      This subroutine calculates the fuel densities at the dose pts.c
c                                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      external dinnerf,dinnera
      common /threel/ dmin,dmax,rhomin,rhomax,rhomean
      common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
      common /thirteen/ numapts
c                                                                        c
c      do integration up to rhomax or a max corresponding to the userc
c      input depending on which is greater                               c
c                                                                        c
      cmax=rhomax
      blah=dlog10(dmax)
      if (cmax.ge.blah) then
        cmax=blah
      end if
      call qromb2(dinnerf,rhomin,cmax,fuel)
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
      function dinnerf(rho)
c                                                                        c
c      This function does the inner integration of equation (1) of       c
c      Jarzemba 1996 while accounting for fuel mass/ash mass to          c
c      calculate the fuel depostion                                      c
c                                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      common /three/ fdmin,fdmax,fdmean,hmin,hmax,xmin,xmax
      external dintegrandf
```

26/69

```
c                                                                       c
c         dintegrandf is the integrand of Eq 1 of Jarzemba (1996)       c
c         including the fuel fraction                                   c
c                                                                       c
c         call qromb(dintegrandf,hmin,hmax,s,rho)
         call qgaus(dintegrandf,hmin,hmax,s,rho)
         dinnerf=s
         return
         end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
         function dinnera(rho)
c                                                                       c
c         This function does the inner integration of equation (1) of   c
c         Jarzemba 1996 to calculate the ash deposition                 c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         implicit double precision (a-h,o-z)
         implicit integer (i-n)
         common /three/ fdmin,fdmax,fdmean,hmin,hmax,xmin,xmax
         external dintegranda
c                                                                       c
c         dintegranda is the integrand of Eq 1 of Jarzemba (1996)       c
c                                                                       c
c         call qromb(dintegranda,hmin,hmax,s,rho)
         call qgaus(dintegranda,hmin,hmax,s,rho)
         dinnera=s
         return
         end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
         function dintegrandf(z,rho)
c                                                                       c
c         This function is the integrand for fuel deposition-           c
c         the coordinate system has to be adjusted to match the main    c
c         coordinate system                                             c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
         implicit double precision (a-h,o-z)
         implicit integer (i-n)
         common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
         common /two/ h,werupt0,airden,airvis,c,u
         common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
         pi=dacos(-1.d0)
c                                                                       c
c         must perfrom a coordinate transformation from the main system c
c         [x,y] to the plume calculation system [xp,yp] that has x      c
c         aligned with the wind                                         c
c                                                                       c
         if (x.ne.0.d0) then
           theta=datan(y/x)
         else if (x.eq.0.d0 .and. y.lt.0.d0) then
           theta=-pi/2.d0
         else if (x.eq.0.d0 .and. y.gt.0.d0) then
           theta=pi/2.d0
         end if
         theta=theta*180.d0/pi
         if (y.lt.0.d0 .and. x.lt.0.d0) then
           theta=theta-180.d0
         else if (y.gt.0.d0 .and. x.lt.0.d0) then
           theta=theta+180.d0
```

27/69

```
          else if (y.eq.0.d0 .and. x.lt.0.d0) then
            theta=-180.d0
          else if (y.eq.0.d0 .and. x.gt.0.d0) then
            theta=0.d0
          end if
          thetap=theta-udir
          if (thetap.ge.-360.d0 .and. thetap.le.-180.d0) then
            thetap=thetap+360.d0
          else if (thetap.gt.180.d0 .and. thetap.le.360.d0) then
            thetap=thetap-360.d0
          end if
          thetapr=thetap*pi/180.d0
          if (thetap.gt.0.d0 .and. thetap.lt.90.d0) then
            xp=dsqrt((x**2+y**2)/(1+(dtan(thetapr))**2))
            yp=dsqrt((x**2+y**2)/(1+(1/(dtan(thetapr))**2)))
          else if (thetap.gt.90.d0 .and. thetap.lt.180.d0) then
            xp=-dsqrt((x**2+y**2)/(1+(dtan(thetapr))**2))
            yp=dsqrt((x**2+y**2)/(1+(1/(dtan(thetapr))**2)))
          else if (thetap.gt.-90.d0 .and. thetap.lt.0.d0) then
            xp=dsqrt((x**2+y**2)/(1+(dtan(thetapr))**2))
            yp=-dsqrt((x**2+y**2)/(1+(1/(dtan(thetapr))**2)))
          else if (thetap.gt.-180.d0 .and. thetap.lt.-90.d0) then
            xp=-dsqrt((x**2+y**2)/(1+(dtan(thetapr))**2))
            yp=-dsqrt((x**2+y**2)/(1+(1/(dtan(thetapr))**2)))
          else if (thetap.eq.0.d0) then
            xp=dsqrt(x**2+y**2)
            yp=0.d0
          else if (thetap.eq.180.d0.or.thetap.eq.-180.d0) then
            xp=-dsqrt(x**2+y**2)
            yp=0.d0
          else if (thetap.eq.90.d0) then
            xp=0.d0
            yp=dsqrt(x**2+y**2)
          else if (thetap.eq.-90.d0) then
            xp=0.d0
            yp=-dsqrt(x**2+y**2)
          end if
          xpcm=1.0d5*xp
          ypcm=1.0d5*yp
          arg=-5.d0*((xpcm-u*tf(z,rho))**2+ypcm**2)/
     &                    (8.d0*c*(tf(z,rho)+ts(z))**(2.5d0))
          dintegrandf=FF(rho)*(5.d0*P(z,rho)*q*f(rho))/
     &                    (8.d0*pi*c*(tf(z,rho)+ts(z))**(2.5d0))*
     &                              dmyexp(arg)
          return
          end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
          function dintegranda(z,rho)
c                                                                      c
c          This function is the integrand for ash deposition-          c
c          the coordinate system has to be adjusted to match the main  c
c          coordinate system                                           c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
          implicit double precision (a-h,o-z)
          implicit integer (i-n)
          common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
          common /two/ h,werupt0,airden,airvis,c,u
          common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
          pi=dacos(-1.d0)
c                                                                      c
c          must perfrom a coordinate transformation from the main system c
```

28/69

```
      function dmyexp(x)
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      if (x.lt.-230.d0) then
        dmyexp=0.d0
      else if (x.gt.230.d0) then
        dmyexp=1.d99
        write(6,*) 'large exponential has been truncated to 1.d99'
      else
        dmyexp=dexp(x)
      end if
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      function tf(z,rho)
c                                                                       c
c      This function subroutine calculates the particle fall time      c
c      in s as a function of height (z) and rho                        c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      tf=0.752d06*((1-dmyexp(-0.0625d0*z))/v0(rho))**0.926d0
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      function ts(z)
c
c      This function subroutine calculates the particle diffusion      c
c      time in the eruption column in s                                c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      common /two/ h,werupt0,airden,airvis,c,u
      zcm=z*1.0d5
      ts=((5.0d0*zcm**2)/(288.d0*c))**(0.4d0)
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      function f(rho)
c                                                                       c
c      This function in the pdf of ash mass with the log-particle      c
c      diameter                                                        c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
      common /threel/ dmin,dmax,rhomin,rhomax,rhomean
      pi=dacos(-1.d0)
      arg=-((rho-rhomean)**2)/(2*dsigma**2)
      f=1/(dsqrt(2*pi)*dsigma)*dmyexp(arg)
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      function dm(rho)
```

29/69

```
c                                                                       c
c         This function is the pdf for fuel mass as a function          c
c         of the log-particle diameter                                  c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         implicit double precision (a-h,o-z)
         implicit integer (i-n)
         common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
         dk1=2.d0/((frhomax-frhomin)*(frhomean-frhomin))
         dk2=-2.d0/((frhomax-frhomin)*(frhomax-frhomean))
         if (rho.le.frhomin.or.rho.ge.frhomax) then
           dm=0.d0
         end if
         if (rho.gt.frhomin.and.rho.lt.frhomean) then
           dm=dk1*(rho-frhomin)
         end if
         if (rho.ge.frhomean.and.rho.lt.frhomax) then
           dm=dk2*(rho-frhomean)+dk1*(frhomean-frhomin)
        end if
         return
         end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
         function FF(rho)
c                                                                       c
c         This function calculates the fuel fraction as a function      c
c         of log-diameter                                               c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         implicit double precision (a-h,o-z)
         implicit integer (i-n)
         external Fcumm,FFintegrand
         common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
         common /seven/ Uran
         common /eight/ rhocut
         cmin=frhomin+rhocut
         call qromb3(FFintegrand,cmin,rho,s)
         FF=s
c                                                                       c
c         If FF is greater than 1 then it is truncated to zero in       c
c         order to remove the contamination that these particles carry  c
c         from the scenario, as they would be too dense to transport by c
c         this mechanism                                                c
c                                                                       c
         if (FF.ge.1.d0) then
           FF=0.d0
         end if
         return
         end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
         function FFintegrand(rho)
c                                                                       c
c         This function is the integrand of Equation 2-7 from           c
c         IM 5708-771-610                                               c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         implicit double precision (a-h,o-z)
         implicit integer (i-n)
         external Fcumm
         common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
         common /seven/ Uran
         common /eight/ rhocut
```

30/65

```
      FFintegrand=Uran/q*dm(rho-rhocut)/(1.d0-Fcumm(rho))
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      function Fcumm(rho)
c                                                                      c
c      This subroutine uses a table lookup to yield the CDF            c
c      from the standard normal distribution                           c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
      common /three1/ dmin,dmax,rhomin,rhomax,rhomean
      dimension fc(400)
c                                                                      c
c      This subroutine will accurately sample from the                 c
c      cdf of the standard normal distribution out to                  c
c      at least the fourth decimal place                               c
c                                                                      c
c                                                                      c
      data (fc(i),i=1,95)
     & /0.0000d0,0.0040d0,0.0080d0,0.0120d0,0.0160d0,
     &  0.0199d0,0.0239d0,0.0279d0,0.0319d0,0.0359d0,
     &  0.0398d0,0.0438d0,0.0478d0,0.0517d0,0.0557d0,
     &  0.0596d0,0.0636d0,0.0675d0,0.0714d0,0.0754d0,
     &  0.0793d0,0.0832d0,0.0871d0,0.0910d0,0.0948d0,
     &  0.0987d0,0.1026d0,0.1064d0,0.1103d0,0.1141d0,
     &  0.1179d0,0.1217d0,0.1255d0,0.1293d0,0.1331d0,
     &  0.1368d0,0.1406d0,0.1443d0,0.1480d0,0.1517d0,
     &  0.1554d0,0.1591d0,0.1628d0,0.1664d0,0.1700d0,
     &  0.1736d0,0.1772d0,0.1808d0,0.1844d0,0.1879d0,
     &  0.1915d0,0.1950d0,0.1985d0,0.2019d0,0.2054d0,
     &  0.2088d0,0.2123d0,0.2157d0,0.2190d0,0.2224d0,
     &  0.2258d0,0.2291d0,0.2324d0,0.2357d0,0.2389d0,
     &  0.2422d0,0.2454d0,0.2486d0,0.2518d0,0.2549d0,
     &  0.2580d0,0.2612d0,0.2642d0,0.2673d0,0.2704d0,
     &  0.2734d0,0.2764d0,0.2794d0,0.2823d0,0.2852d0,
     &  0.2881d0,0.2910d0,0.2939d0,0.2967d0,0.2996d0,
     &  0.3023d0,0.3051d0,0.3078d0,0.3106d0,0.3133d0,
     &  0.3159d0,0.3186d0,0.3212d0,0.3238d0,0.3264d0/
      data (fc(i),i=96,190)
     & /0.3289d0,0.3315d0,0.3340d0,0.3365d0,0.3389d0,
     &  0.3413d0,0.3438d0,0.3461d0,0.3485d0,0.3508d0,
     &  0.3531d0,0.3554d0,0.3577d0,0.3599d0,0.3621d0,
     &  0.3643d0,0.3665d0,0.3686d0,0.3708d0,0.3729d0,
     &  0.3749d0,0.3770d0,0.3790d0,0.3810d0,0.3830d0,
     &  0.3849d0,0.3869d0,0.3888d0,0.3907d0,0.3925d0,
     &  0.3944d0,0.3962d0,0.3980d0,0.3997d0,0.4015d0,
     &  0.4032d0,0.4049d0,0.4066d0,0.4082d0,0.4099d0,
     &  0.4115d0,0.4131d0,0.4147d0,0.4162d0,0.4177d0,
     &  0.4192d0,0.4207d0,0.4222d0,0.4236d0,0.4251d0,
     &  0.4265d0,0.4279d0,0.4292d0,0.4306d0,0.4319d0,
     &  0.4332d0,0.4345d0,0.4357d0,0.4370d0,0.4382d0,
     &  0.4394d0,0.4406d0,0.4418d0,0.4429d0,0.4441d0,
     &  0.4452d0,0.4463d0,0.4474d0,0.4484d0,0.4495d0,
     &  0.4505d0,0.4515d0,0.4525d0,0.4535d0,0.4545d0,
     &  0.4554d0,0.4564d0,0.4573d0,0.4582d0,0.4591d0,
     &  0.4599d0,0.4608d0,0.4616d0,0.4625d0,0.4633d0,
     &  0.4641d0,0.4649d0,0.4656d0,0.4664d0,0.4671d0,
     &  0.4678d0,0.4686d0,0.4693d0,0.4699d0,0.4706d0/
      data (fc(i),i=191,285)
```

31/69

```
          Fcumm=0.5d0
        end if
         return
         end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
         function P(z,rho)
c                                                                       c
c         This function subroutine calculates the probability           c
c         density function as a function of height for                 c
c         a given particle size                                        c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         implicit double precision (a-h,o-z)
         implicit integer (i-n)
         common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
         common /two/ h,werupt0,airden,airvis,c,u
         common /three/ fdmin,fdmax,fdmean,hmin,hmax,xmin,xmax
c
c         The formulation of P(z) in Suzuki (1983) is incorrect. The
c         correct formulation should have Y(z)=Beta*W(z)/V0. This
c         subroutine has been corrected.
c         MSJ 5/7/97
c
         wz=werupt0*(1-z/hmax)
c         yz=beta*(wz-v0(rho))/v0(rho)
        yz=beta*wz/v0(rho)
c         y0=beta*(werupt0-v0(rho))/v0(rho0
        y0=beta*werupt0/v0(rho)
         anum= beta*werupt0*yz*dmyexp(-yz)
         denom= v0(rho)*hmax*(1.d0-(1.d0+y0)*dmyexp(-y0))
         P=anum/denom
c                                                                       c
c         The formula above, directly from Suzuki (1983) has the       c
c         possiblity of being negative for certain values of z.        c
c         Must set to zero if this is ever so.                         c
c                                                                       c
c         With the correction added above, this comment is no longer   c
c         accurate MSJ 5/7/97                                          c
c                                                                       c
         if (P.le.0.d0) then
           P=0.d0
         end if
              return
         end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
         function v0(rho)
c                                                                       c
c         The argument d is the diameter of the particle in cm         c
c         v0 is the terminal fall velocity of particles at sea-level   c
c         given in Suzuki, 1983.                                       c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c         g : gravitational accelleration in cm/s**2
c
         implicit double precision (a-h,o-z)
         implicit integer (i-n)
         common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
         common /two/ h,werupt0,airden,airvis,c,u
         g=980.d0
         d=10.d0**rho
```

32/69

```
      term1=9.d0*airvis*fshape**(-0.32d0)
      term2=81.d0*airvis**2*fshape**(-.64d0)
      term3=1.5d0*airden*ashden(rho)*d**3*g*dsqrt(1.07d0-fshape)
      anum = ashden(rho)*g*d**2
      denom = term1+dsqrt(term2 + term3)
      v0 = anum/denom
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      function ashden(rho)
c                                                                      c
c      This function calculates the contaminated ash density           c
c      as a function of rho                                            c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      external f,FF
      common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
      common /eleven/ ashrholow,ashrhohi
      if (rho.ge.ashrhohi) then
        ashden=ashdenmin
      else if (rho.le.ashrholow) then
        ashden=ashdenmax
      else
        slope=(ashdenmin-ashdenmax)/(ashrhohi-ashrholow)
        yint=ashdenmax
        ashden=slope*(rho-ashrholow)+yint
      end if
c                                                                      c
c      Need to correct for any fuel that has been absorbed             c
c      by the ash                                                      c
c                                                                      c
      ashden=ashden*(1.d0+FF(rho))
     return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      subroutine inputdata
c                                                                      c
c      This subroutine reads the input data and also stochastically    c
c      calculates the important parameters                             c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      parameter (nmx=200, nmy=200)
      parameter (maxd= 200)
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      character*60 title
      common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
      common /two/ h,werupt0,airden,airvis,c,u
      common /three/ fdmin,fdmax,fdmean,hmin,hmax,xmin,xmax
      common /three1/ dmin,dmax,rhomin,rhomax,rhomean
      common /four/ ymin,ymax,acutoff
      common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
      common /six/ numptsx,numptsy
      common /seven/ Uran
      common /eight/ rhocut
      common /nine/ v,icount
      common /eleven/ ashrholow,ashrhohi
      common /twelve/ power,tdur
      common /sixteen/ title
```

33/67

```
      qdot=(hmax/0.24)**4.d0
      q=qdot*tdur*1.0d3
c                                                                       c
c      wind direction-quasi-uniform using SCP 5,000 ft data             c
c                                                                       c
      call winddirect(udir)
c                                                                       c
c      wind speed-exponential with direction dependent lambda           c
c                                                                       c
      call windspeed (udir,u)
c                                                                       c
c      dmean-logtriangular                                              c
c      dsigma-loguniform                                                c
c                                                                       c
      rnd5=v(icount)
      icount=icount+1
      call triangular(rhomean,dmeanmin,dmeanmed,dmeanmax)
           dmean=10.d0**rhomean
c                                                                       c
c     dsigma is sampled uniformly between min and max                   c
c                                                                       c
           dsigma= dsigmamin+rnd5*(dsigmamax-dsigmamin)
c                                                                       c
c      beta-loguniform                                                  c
c                                                                       c
           rnd6=v(icount)
      icount=icount+1
      betalog=betalogmin+(betalogmax-betalogmin)*rnd6
      beta=10.d0**betalog
      rhomin=rhomean-5.d0*dsigma
      rhomax=rhomean+5.d0*dsigma
      rlog=-2.069d0+2.d0*dlog10(ashden(rhomean))+0.27435d0*
     &                 dlog10(qdot*1.d3)
      vrad=10.d0**rlog
      werupt0=qdot*1.d3/(ashden(rhomean)*3.141592d0*(vrad*1.d2)**2)
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      subroutine histf(rhomin,cmax,x,y,ivol,frhomin,frhomax,frhomean)
c                                                                       c
c      This subroutine is "in charge" of calculating the fuel           c
c      particulate size distribution at the dose point                  c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      character*19 out
      character*4 ixx,iyy,ivv
      external FFintegrandhist
     common /eight/ rhocut
      common /thirteen/ numapts
      ixloc=int(x)
      iyloc=int(y)
      write(ixx,300) ixloc
      write(iyy,300) iyloc
      write(ivv,300) ivol
300      format(i4)
      call filename('fuel',x,y,ivol,out)
      open(unit=22,file=out,status='unknown')
      write(22,301)'rho-f','fuel mass (g)'
301      format(a12,a15)
      do ii=1,numapts+1
```

34/69

```
          rhof=frhomin+dfloat(ii-1)/(numapts-1)*(frhomax-frhomin)
          dmina=rhof+rhocut
          if (dmina.lt.rhomin) then
            dmina=rhomin
          end if
          call qromb4(FFintegrandhist,dmina,cmax,ans,rhof,x,y,ivol)
          blah=rhof-rhocut
          write(22,302) blah,ans
302          format(f12.4,e15.5)
        end do
        close (unit=22)
        return
        end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
        function FFintegrandhist(rhoa,rhof,x,y,ivol)
c                                                                      c
c         This function is the integrand for calculating particle size c
c         distributions at the dose points                             c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        implicit double precision (a-h,o-z)
        implicit integer (i-n)
        common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
        common /seven/ Uran
        common /eight/ rhocut
        external dma
        if (rhoa.ge.rhof+rhocut) then
          FFintegrandhist=Uran/q*dm(rhof)/
     &          (1.d0-Fcumm(rhof+rhocut))*dma(rhoa,x,y,ivol)
      else
          FFintegrandhist=0.d0
        end if
        return
        end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
        function dma(rhoa,x,y,ivol)
c                                                                      c
c         This function reads in the ash deposition at the dose point  c
c         currently being examined                                     c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        implicit double precision (a-h,o-z)
        implicit integer (i-n)
        character*19 infile
      character*12 blah1
        character*15 blah2
        character*4 ix,iy,iv
        dimension dr(31),dd(31)
        common /thirteen/ numapts
        ixloc=int(x)
        iyloc=int(y)
        write(ix,300) ixloc
        write(iy,300) iyloc
        write(iv,300) ivol
300        format(i4)
        call filename('ashh',x,y,ivol,infile)
        open (unit=30,file=infile,status='unknown')
        read (30,299) blah1,blah2
299        format (a12,a15)
        do i=1,numapts
          read(30,301) dr(i),dd(i)
```

35/69

```
     &                    -112.5d0,-135.d0,-157.5d0,1000.d0/
          rnd=v(icount)
          icount=icount+1
          do i=1,17
                   if (rnd.le.cutoff(i)) then
                            udir=angle(i)
                            go to 10
                   end if
          end do
c                                                                       c
c          This statement sets the wind direction to -90 degrees        c
c          for all realizations                                         c
c                                                                       c
10       continue
         return
          end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
          subroutine windspeed(udir,u)
c                                                                       c
c          This subroutine selects the wind speed u given a             c
c          wind direction udir as described                             c
c          in Jarzemba (1996)                                           c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c
c
          implicit double precision (a-h,o-z)
          implicit integer (i-n)
          common /nine/ v,icount
          dimension angle(17),ula(17),v(10000)
          data angle/0.d0,22.5d0,45.d0,67.5d0,90.d0,112.5d0,135.d0,
     &                    157.5d0,180.d0,-22.5d0,-45.d0,-67.5d0,-90.d0,
     &                    -112.5d0,-135.d0,-157.5d0,1000.d0/
               data ula/0.313d0,0.222d0,0.149d0,0.139d0,0.156d0,
     &                    0.217d0,0.323d0,0.4d0,0.417d0,0.244d0,
     &                    0.213d0,0.189d0,0.172d0,0.185d0,0.208d0,
     &                    0.294d0,99999.d0/
          do i=1,17
                   if (udir.eq.angle(i)) then
                            ulambda=ula(i)
                   end if
          end do
          if (udir.eq.1000.d0) then
                   udir=0.d0
                   u=0.d0
          end if
          rnd=v(icount)
          icount=icount+1
          u=-dlog(1-rnd)/ulambda
          u=u*100.d0
          return
          end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
          subroutine qromb4(func,a,b,s,rhof,x,y,ivol)
c                                                                       c
c          This subroutine is used to calculate the fuel particulate    c
c          distribution with size at the dose point.                    c
c          It is origially from Numerical Recipes.                      c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

36/69

```
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      parameter (eps=30.d-4,jmax=22,jmaxp=jmax+1,k=5,km=k-1)
      external func
      dimension ss(jmaxp),h(jmaxp)
      h(1)=1.d0
      it=0
      do j=1,jmax
        call trapzd4(func,a,b,ss(j),j,it,rhof,x,y,ivol)
        if (j.ge.k) then
          call polint4(h(j-km),ss(j-km),k,0.d0,s,dss)
          if (dabs(dss).le.eps*dabs(s)) then
            return
            end if
        end if
        ss(j+1)=ss(j)
        h(j+1)=0.25d0*h(j)
      end do
      write(6,*) 'integration surpassed iteration limit'
      write(6,*) 'for calculating fuel particluate'
      write(6,*) 'distribution at the dose point.'
      write(6,*) 'Accuracy of fuel distribution'
      write(6,*) 'cannot be gauranteed.'
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      subroutine trapzd4(func,a,b,s,n,it,rhof,x,y,ivol)
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      external func
      if (n.eq.1) then
              s=0.5d0*(b-a)*(func(a,rhof,x,y,ivol)+
     &                                func(b,rhof,x,y,ivol))
              it=1
      else
              itnm=it
              del=(b-a)/itnm
              xx=a+0.5d0*del
              sum=0.d0
              do j=1,it
                      sum=sum+func(xx,rhof,x,y,ivol)
                      xx=xx+del
              end do
              s=0.5d0*(s+(b-a)*sum/itnm)
              it=2*it
      end if
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      subroutine polint4(xa,ya,n,x,y,dy)
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      parameter (nmax=10)
      dimension xa(n),ya(n),c(nmax),d(nmax)
      ns=1
      dif=dabs(x-xa(1))
      do i=1,n
```

37/65

```
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      subroutine trapzd3(func,a,b,s,n,it)
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      external func
      if (n.eq.1) then
              s=0.5d0*(b-a)*(func(a)+func(b))
              it=1
      else
              itnm=it
              del=(b-a)/itnm
              xx=a+0.5d0*del
              sum=0.d0
              do j=1,it
                      sum=sum+func(xx)
                      xx=xx+del
              end do
              s=0.5d0*(s+(b-a)*sum/itnm)
              it=2*it
      end if
      return
      end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
      subroutine polint3(xa,ya,n,x,y,dy)
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      parameter (nmax=10)
      dimension xa(n),ya(n),c(nmax),d(nmax)
      ns=1
      dif=dabs(x-xa(1))
      do i=1,n
        dift=dabs(x-xa(i))
        if (dift.lt.dif)then
          ns=i
          dif=dift
        end if
        c(i)=ya(i)
        d(i)=ya(i)
      end do
      y=ya(ns)
      ns=ns-1
      do m=1,n-1
        do i=1,n-m
          ho=xa(i)-x
          hp=xa(i+m)-x
          w=c(i+1)-d(i)
          den=ho-hp
          if (den.eq.0.d0) then
            write(6,*) 'error in polint3'
            return
          end if
          den=w/den
          d(i)=hp*den
          c(i)=ho*den
        end do
        if (2*ns.lt.n-m) then
```

38/65

```
            dy=c(ns+1)
          else
            dy=d(ns)
            ns=ns-1
          end if
          y=y+dy
        end do
        return
        end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
        subroutine qromb2(func,a,b,s)
c                                                                      c
c       This subroutine is used to calculate the outer integral for    c
c       the fuel distribution at the point of interest                 c
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        implicit double precision (a-h,o-z)
        implicit integer (i-n)
c                                                                      c
c          Recommend using eps=1.d-4 MSJ 5/7/97                        c
c                                                                      c
        parameter (eps=1.d-4,jmax=22,jmaxp=jmax+1,k=5,km=k-1)
        external func
        dimension ss(jmaxp),h(jmaxp)
        h(1)=1.d0
        it=0
        do j=1,jmax
          call trapzd2(func,a,b,ss(j),j,it)
          if (j.ge.k) then
            call polint2(h(j-km),ss(j-km),k,0.d0,s,dss)
            if (dabs(dss).le.eps*dabs(s)) then
              return
            end if
          end if
          ss(j+1)=ss(j)
          h(j+1)=0.25d0*h(j)
        end do
        write(6,*) 'Integration surpassed iteration limit'
        write(6,*) 'for calculating outer integral for fuel'
        write(6,*) 'areal density at the dose point.'
        write(6,*) 'Accuracy cannot be gauranteed.'
        return
        end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
        subroutine trapzd2(func,a,b,s,n,it)
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
        implicit double precision (a-h,o-z)
        implicit integer (i-n)
        external func
        if (n.eq.1) then
                s=0.5d0*(b-a)*(func(a)+func(b))
                it=1
        else
                itnm=it
                del=(b-a)/itnm
                xx=a+0.5d0*del
                sum=0.d0
                do j=1,it
                        sum=sum+func(xx)
                        xx=xx+del
```

39/69

```
351           format(a1,i3)
          end if
c                                                                       c
c          Calculate volcano number character string                    c
c                                                                       c
          if(iabs(ivol).ge.0.and.iabs(ivol).lt.10) then
            write(iv,360)'000',iabs(ivol)
360           format(a3,i1)
          else if(iabs(ivol).ge.10.and.iabs(ivol).lt.100) then
            write(iv,370)'00',iabs(ivol)
370           format(a2,i2)
          else if(iabs(ivol).ge.100.and.iabs(ivol).lt.1000) then
            write(iv,380)'0',iabs(ivol)
380           format(a1,i3)
          else if(iabs(ivol).ge.1000.and.iabs(ivol).lt.10000) then
            write(iv,390) ivol
390           format(i4)
          else
            write(6,*)'ivol=',ivol
            write(6,*) 'error in filename'
            pause
            stop
          end if
          file=type//'x'//ix//'y'//iy//'v'//iv
          return
          end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
        subroutine qgaus(func,a,b,ss,rho)
c                                                                       c
c          This subroutine uses n-point gauss-legendre integration      c
c          to do the inner integral for both ash and fuel depositions.  c
c          It uses gauleg to calculate the abscissas and weights        c
c          for the integration.                                         c
c                                                                       c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
         implicit double precision (a-h,o-z)
        implicit integer (i-n)
c                                                                       c
c          n=250 generally gaurantees @1% accuracy                      c
c          n=400 generally gaurantees maximum accuracy                  c
c                                                                       c
c          Recommend using n=20 MSJ 5/7/97                              c
c                                                                       c
         parameter (n=20)
         external func
         dimension x(n),w(n)
        call gauleg(0.d0,1.d0,x,w,n)
        xm=0.5d0*(b+a)
        xr=0.5d0*(b-a)
        ss=0.d0
        do 11 j=1,n
           dx=xr*x(j)
           ss=ss+w(j)*(func(xm+dx,rho)+func(xm-dx,rho))
11      continue
        ss=xr*ss
         return
         end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                       c
        subroutine gauleg(x1,x2,x,w,n)
c                                                                       c
c          This subroutine is called by gauleg to calculate the         c
```

Information on Pages 40 through 48 contains FxDebgugger for Power Macintosh User Guide copyright information and is therefore not included in this file.

1/24/96 CONTINUED INVESTIGATIONS OF EXPOSURES FROM EXTRUSIVE VOLCANISM
MSJ

PURPOSE: TO CONTINUE INVESTIGATIONS, AS ORIGINALLY SHOWN IN JARZEMBA (1995).

PROBABILITY OF VOLCANISM OCCURRING AT YM

CONNOR AND HILL (1995) USE DIFFERING PROBABILITY ANALYSES TO DETERMINE THAT THE PROBABILITY OF A NEW CONE FORMING IN THE REPOSITORY AREA IN THE NEXT 10,000 YEARS IS:

$$1 \times 10^{-4} \text{ TO } 5 \times 10^{-4}$$

WHERE THE REPOSITORY AREA IS THE FOOTPRINT PLUS A 500 m BUFFER ZONE (EQUAL TO ABOUT $8 km^2$)

ASSUMPTION: 1) PROBABILITY IS CONSTANT IN SPACE AND TIME OVER THE REPOSITORY AREA (FOOTPRINT PLUS SEVERAL km BUFFER ZONE).
2) USE MEAN VALUE FROM RANGE ABOVE, 3×10^{-4} $1/10{,}000\,YR \cdot 8\,km^2$

$$\text{RECURRANCE RATE} = RR. = \frac{3 \times 10^{-4}}{10^4\,YR \cdot 8\,km^2} = 3.75 \times 10^{-9}\ 1/YR \cdot km^2$$

NEED TO FIND RR FOR THE AREA SHOWN IN FIGURE 2 FROM THE VOLCANO MODULE.

VOLCANO USES AN AREA W $\Delta Y = 12\,km$, $\Delta X = 12\,km$
OR AREA = $144\ km^2$

PROBABILITY OVER VOLCANO AREA IS:

$$RR_v = 3.75 \times 10^{-9}\ 1/YR \cdot km^2\ (144\,km^2) = \boxed{5.4 \times 10^{-7}\ 1/YR}$$

7-26-96 VERIFICATION EXCERSIZE 50/69

MSJ

COMPARISON OF CODE RESULTS W/ PUBLISHED CALCULATIONS (SUZUKI (1983))

INPUT DATA SET

TOTAL ASH MASS = 10^{15} g EVENT DURATION (T) = 3.3177E5 s

COLUMN HT = 10 km $\dot{Q}$ = 3.014E9 g/s

ERUPTION VELOCITY = 10^4 cm/s

MEAN PARTICLE DIAMETER = 1 mm

SIGMA ON PARTICLE DIAMETER = 0.4

WIND SPEED = 10^3 cm/s

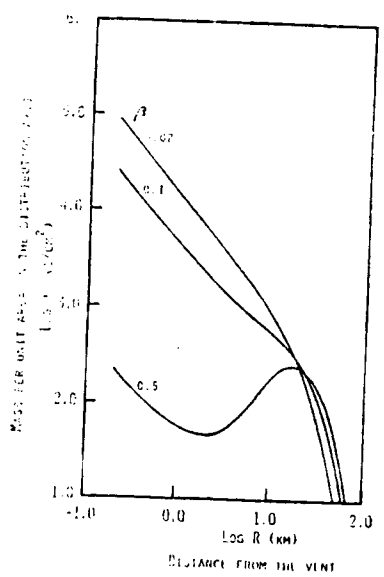
ASH DENSITY = 0.8 g/cm^3

PARTICLE SHAPE PARAMETER = 0.5

AIR DENSITY = 0.001293 g/cc

AIR VISCOSITY = 0.00018 g/cm-s

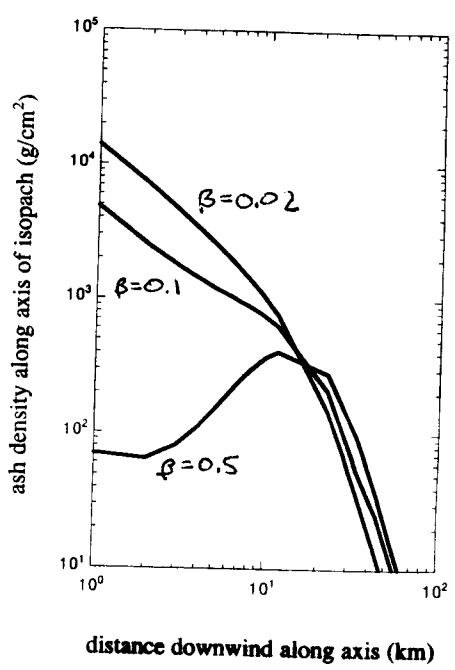
EDDY DIFFUSIVITY = 400 cm^2/s$^{5/2}$



SUZUKI (1983)

FROM ASHPLUME

Comparison plot with Suzuki (1983)



51/69

MARK JARZEMBA

The Boorum & Pease® Quality Guarantee

The materials and craftsmanship that went into this product are of the finest quality. The pages are thread sewn, meaning they're bound to stay bound. The inks are moisture resistant and will not smear. And the uniform quality of the paper assures consistent rulings, excellent writing surface and erasability. If, at any time during normal use, this product does not perform to your expectations, we will replace it free of charge. Simply write to us:

Boorum & Pease Company
71 Clinton Road, Garden City, NY 11530
Attn: Marketing Services

Any correspondence should include the code number printed at the bottom of this page as well as the book title stamped at the bottom of the spine.

CNWRA
CONTROLLED
COPY 164

One Good Book Deserves Many Others.

Look for the complete line of Boorum & Pease® Columnar, Journal, and Record books. Custom-designed books also available by special order. For more information about our Customized Book Program, contact your office products dealer. See back cover for other books in this series.

Made in U.S.A.
RMI171193

SOFTWARE PROBLEM/CHANGE REPORT

01. SPCR Number: PA-SPCR-102

02. Project Title: TSPA&I Code Development | 20-5708-762

03. SPCR Title: Change of a units conversion error in ASHRMOVO Module and Other Changes of Lesser Importance

04. AFFECTED SOFTWARE AND/OR DOCUMENTATION

1. ashrmovo.f

05. DESCRIPTION OF PROBLEM/CHANGE

1. Page 11, lines 10-11 of the attachment: The units conversion constant at the end of this line (1.0d-4) incorrectly converts Ci/cm^2 to Ci/m^2. The correct conversion constant should be 1.0d+4. This change should be made by commenting out lines 51 and 52, and replacing them with lines 49 and 50 as shown on the attachment.

2. Page 3, Line 51 to Page 5 Line 35:The labeling of all Kd values should be in 'cm3/g'. This change has been made on the attached version. It is noted that the actual values were always in cm^3/g they were just mislabeled and that the parameter labels in the input file should be changed accordingly.

3. Page 10 Line 24: The argument of the if statement has been changed due to one of the terms being incomplete (see attachment).

4. Page10 Lines 33-34: Similar to 3. above, this line has been changed to complete the last term in the equation.

06. PROBLEM SOLUTION/NEED FOR CHANGE

1. This error results in the underprediction of doses from the extrusive volcanism scenario by eight orders of magnitude.

2.-4. These errors are of lesser importance but should still be changed to the corrected solutions in the attachment. They only affect the point at which the release has become rate limited as opposed to solubility limited.

07. Originator: Mark Jarzemba	Title: Research Engineer	Date: 5/27/97

PROJECT

08. Need by Date:6/1/97	09. Approved: RGB	Disapproved:

10. Software Developer: S. Mohanty et. al.	Date: 5/27/97
11. Element Manager: RG Baca	Date: 6/5/97

12. IMPLEMENTED SOLUTION

See the attachment as itemized above.

13. Implemented By: Mark Jarzemba	Date: 6-4-97

CNWRA Form TOP-5 (03/95)

53/69

postscript

JOB 783

MSJ:TPA:test_files:ashplume_tests:ash plume.in

For:	**MSJ**
Creator:	**MPW Shell: LaserWriter 8 8.1.1**
Creation Date:	**9:13 AM Friday, June 6, 1997**
Submit queue:	IF 1 / Ethernet / ETHERTALK
Submitted:	Sun Apr 13 01:11:23 1992
Started:	Sun Apr 13 01:11:24 1992

QMS 3825 Print System

54/69

```
No Title
-40.d0  40.d0                    ! xmin, xmax in km
-100.d0  6.d0                    ! ymin, ymax in km
41                               ! numptsx
54                               ! numptsy
4.80d0  6.86d0                   ! tlogmin, tlogmax- logs of t in sec
9.41    11.55                    ! powlogmin, powlogmax- logs of P in W
-2.0d0  -0.3d0                   ! betalogmin, betalogmax-logs
-2.0d0  -1.0d0  1.0d0            ! dmeanmin, dmeanmed, dmeanmax-logs of d in cm
0.1d0   1.0d0                    ! dsigmamin, dsigmamax
0.8     0.8                      ! ashdenmin, ashdenmax in g/cm3
-2.d0   -1.d0                    ! ashrholow, ashrhohi
0.5d0                            ! fshape
0.001293d0      1.8d-04          ! airden in g/cm3, airvis in g/cm-s
400.d0                           ! c in cm2/s to the 5/2
10.d0                            ! dmax in cm
0.0001d0 0.001d0 0.01d0          ! fdmin, fdmean, fdmax al in cm
0.001d0                          ! hmin in km
1.d-10d0                         ! acutoff in g/cm2
0.3d0                            ! rhocut-incorporation ratio
1.d8                             ! Uran- total mass of fuel in g
```

55/69

postscript

JOB 782

MSJ:TPA:test_files:ashplume_tests:ashplume.out

For:	**MSJ**
Creator:	**MPW Shell: LaserWriter 8 8.1.1**
Creation Date:	**9:11 AM Friday, June 6, 1997**
Submit queue:	IF 1 / Ethernet / ETHERTALK
Submitted:	Sun Apr 13 01:08:55 1992
Started:	Sun Apr 13 01:08:56 1992

QMS

QMS 3825 Print System

QMS 3825 Print System

56/69

```
 iseed=            10
 ASHPLUME version 1.0
No Title

*********************************************************************
*                                                                   *
*                     realization number                    1       *
*                      wind speed (cm/s)            1000.0000       *
*                   wind direction (deg)             -90.0000       *
*            mean particle diameter (cm)               0.1000       *
*                           log- std dev               0.4000       *
*                       column ht (km)                10.0000       *
*                    event duration (s)           0.3318E+06       *
*                          ash mass (g)           0.1000E+16       *
*                       event power (W)           0.2212E+13       *
*                                 beta                0.1000       *
*              vent exit velocity (cm/s)           10000.0000       *
*               particle shape parameter               0.5000       *
*                    air density (g/cc)           0.1293E-02       *
*                 air viscosity (g/cm-s)           0.1800E-03       *
*         eddy diff. constant (cm2/s5/2)             400.0000       *
*                       size cutoff (cm)              10.0000       *
*                     incorporation ratio              0.3000       *
*         fuel particle minimum log-diam              -4.0000       *
*          fuel particle median log-diam              -3.0000       *
*         fuel particle maximum log-diam              -2.0000       *
*          total fuel mass available (g)           0.1000E+09       *
*                                                                   *
*********************************************************************
```

x (km)	y (km)	xash (g/cm^2)	xfuel (g/cm^2)
-40.000	-100.000	0.1293E+00	0.1287E-07
-40.000	-98.000	0.1287E+00	0.1281E-07
-40.000	-96.000	0.1276E+00	0.1271E-07
-40.000	-94.000	0.1262E+00	0.1258E-07
-40.000	-92.000	0.1244E+00	0.1240E-07
-40.000	-90.000	0.1222E+00	0.1218E-07
-40.000	-88.000	0.1195E+00	0.1192E-07
-40.000	-86.000	0.1164E+00	0.1161E-07
-40.000	-84.000	0.1129E+00	0.1126E-07
-40.000	-82.000	0.1089E+00	0.1086E-07
-40.000	-80.000	0.1045E+00	0.1043E-07
-40.000	-78.000	0.9972E-01	0.9953E-08
-40.000	-76.000	0.9456E-01	0.9439E-08
-40.000	-74.000	0.8907E-01	0.8892E-08
-40.000	-72.000	0.8329E-01	0.8315E-08
-40.000	-70.000	0.7727E-01	0.7715E-08
-40.000	-68.000	0.7108E-01	0.7097E-08
-40.000	-66.000	0.6478E-01	0.6469E-08
-40.000	-64.000	0.5851E-01	0.5841E-08
-40.000	-62.000	0.5220E-01	0.5212E-08
-40.000	-60.000	0.4608E-01	0.4601E-08
-40.000	-58.000	0.4018E-01	0.4011E-08
-40.000	-56.000	0.3458E-01	0.3452E-08
-40.000	-54.000	0.2934E-01	0.2929E-08
-40.000	-52.000	0.2453E-01	0.2448E-08
-40.000	-50.000	0.2018E-01	0.2014E-08
-40.000	-48.000	0.1633E-01	0.1630E-08
-40.000	-46.000	0.1299E-01	0.1296E-08
-40.000	-44.000	0.1014E-01	0.1011E-08
-40.000	-42.000	0.7770E-02	0.7748E-09
-40.000	-40.000	0.5839E-02	0.5821E-09

57/69

-40.000	-38.000	0.4303E-02	0.4289E-09
-40.000	-36.000	0.3111E-02	0.3099E-09
-40.000	-34.000	0.2206E-02	0.2198E-09
-40.000	-32.000	0.1538E-02	0.1531E-09
-40.000	-30.000	0.1091E-02	0.1080E-09
-40.000	-28.000	0.7319E-03	0.7247E-10
-40.000	-26.000	0.4776E-03	0.4730E-10
-40.000	-24.000	0.3185E-03	0.3150E-10
-40.000	-22.000	0.2125E-03	0.2096E-10
-40.000	-20.000	0.1426E-03	0.1403E-10
-40.000	-18.000	0.9691E-04	0.9495E-11
-40.000	-16.000	0.6710E-04	0.6541E-11
-40.000	-14.000	0.4757E-04	0.4611E-11
-40.000	-12.000	0.3466E-04	0.3337E-11
-40.000	-10.000	0.2600E-04	0.2485E-11
-40.000	-8.000	0.2008E-04	0.1904E-11
-40.000	-6.000	0.1595E-04	0.1512E-11
-40.000	-4.000	0.1295E-04	0.1210E-11
-40.000	-2.000	0.1076E-04	0.9978E-12
-40.000	0.000	0.9111E-05	0.8390E-12
-40.000	2.000	0.7837E-05	0.7171E-12
-40.000	4.000	0.6833E-05	0.6215E-12
-40.000	6.000	0.6026E-05	0.5450E-12
-38.000	-100.000	0.1547E+00	0.1540E-07
-38.000	-98.000	0.1548E+00	0.1542E-07
-38.000	-96.000	0.1545E+00	0.1540E-07
-38.000	-94.000	0.1538E+00	0.1533E-07
-38.000	-92.000	0.1526E+00	0.1522E-07
-38.000	-90.000	0.1510E+00	0.1505E-07
-38.000	-88.000	0.1488E+00	0.1484E-07
-38.000	-86.000	0.1460E+00	0.1457E-07
-38.000	-84.000	0.1427E+00	0.1424E-07
-38.000	-82.000	0.1388E+00	0.1385E-07
-38.000	-80.000	0.1344E+00	0.1341E-07
-38.000	-78.000	0.1294E+00	0.1291E-07
-38.000	-76.000	0.1238E+00	0.1236E-07
-38.000	-74.000	0.1177E+00	0.1175E-07
-38.000	-72.000	0.1111E+00	0.1110E-07
-38.000	-70.000	0.1042E+00	0.1040E-07
-38.000	-68.000	0.9704E-01	0.9689E-08
-38.000	-66.000	0.8908E-01	0.8894E-08
-38.000	-64.000	0.8132E-01	0.8120E-08
-38.000	-62.000	0.7879E-01	0.7860E-08
-38.000	-60.000	0.6549E-01	0.6539E-08
-38.000	-58.000	0.5773E-01	0.5764E-08
-38.000	-56.000	0.5022E-01	0.5013E-08
-38.000	-54.000	0.4307E-01	0.4299E-08
-38.000	-52.000	0.3638E-01	0.3630E-08
-38.000	-50.000	0.3023E-01	0.3016E-08
-38.000	-48.000	0.2469E-01	0.2463E-08
-38.000	-46.000	0.1979E-01	0.1974E-08
-38.000	-44.000	0.1556E-01	0.1552E-08
-38.000	-42.000	0.1199E-01	0.1196E-08
-38.000	-40.000	0.9048E-02	0.9020E-09
-38.000	-38.000	0.6680E-02	0.6658E-09
-38.000	-36.000	0.4825E-02	0.4807E-09
-38.000	-34.000	0.3409E-02	0.3396E-09
-38.000	-32.000	0.2358E-02	0.2349E-09
-38.000	-30.000	0.1600E-02	0.1592E-09
-38.000	-28.000	0.1066E-02	0.1061E-09
-38.000	-26.000	0.7000E-03	0.6730E-10
-38.000	-24.000	0.4637E-03	0.4582E-10

58/69

postscript

JOB 780

MSJ:TPA:test_files:ashplume_tests:integrator.f

For:	**MSJ**
Creator:	**MPW Shell: LaserWriter 8 8.1.1**
Creation Date:	**9:10 AM Friday, June 6, 1997**
Submit queue:	IF 1 / Ethernet / ETHERTALK
Submitted:	Sun Apr 13 01:08:03 1992
Started:	Sun Apr 13 01:08:04 1992

QMS 3825 Print System

59/65

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                      c
      program integrator
c                                                                      c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      character*50 char
      open (unit=12,file='ashplume.in',status='old')
      open (unit=8,file='ashplume.out',status='old')
      open (unit=10,file='integrator.out',status='unknown')
      read(12,*)
      read(12,*) xmin,xmax
      read(12,*) ymin,ymax
      read(12,*) numptsx
      read(12,*) numptsy
      write(10,130) '   xmin= ',xmin,'    xmax= ',xmax
      write(10,130) '   ymin= ',ymin,'    ymax= ',ymax
      write(10,120) 'numptsx= ', numptsx,' numptsy= ',numptsy
120   format(a9,i5,a10,i5)
130   format(a9,f8.3,a10,f8.3)
      close (unit=12)
      do i=1,13
       read(8,*)
      end do
      read(8,100) char,ashmass
      do i=1,12
       read(8,*)
      end do
      read(8,100) char,fuelmass
      do i=1,4
       read(8,*)
      end do
100   format(a50,e10.4)
      dint=0.d0
      dintf=0.d0
      deltax=dabs(xmax-xmin)/(numptsx-1)
      deltay=dabs(ymax-ymin)/(numptsy-1)
      deltaxcm=deltax*1.d5
      deltaycm=deltay*1.d5
      write(10,140)'deltaxcm= ',deltaxcm,' deltaycm= ',deltaycm
140   format(a10,f10.3,a11,f10.3)
      do j=1,numptsx
       read(8,*)
       do k=1,numptsy
         read(8,*) x,y,xash,xfuel
         if(xash.lt.0.0d0.or.xfuel.lt.0.0d0) then
           print *,'error: negative deposition'
           pause
         end if
         dint=dint+xash*deltaxcm*deltaycm
         dintf=dintf+xfuel*deltaxcm*deltaycm
       end do
      end do
      write(10,110) 'ash mass integrated over isopach (g) = ',dint
      write(10,110) 'ash mass extruded in event (g) = ',ashmass
      write(10,110) 'fuel mass integrated over isopach (g) = ',dintf
      write(10,110) 'fuel mass extruded in event (g) = ',fuelmass
110   format(a50,e10.4)
      close (unit=8)
      close (unit=10)
      end
```

60/69

postscript

JOB 781

MSJ:TPA:test_files:ashplume_tests:integrator.out

For: MSJ
Creator: MPW Shell: LaserWriter 8 8.1.1
Creation Date: 9:11 AM Friday, June 6, 1997

Submit queue: IF 1 / Ethernet / ETHERTALK
Submitted: Sun Apr 13 01:08:35 1992
Started: Sun Apr 13 01:08:42 1992

QMS 3825 Print System

61/69

```
   xmin=  -40.000    xmax=   40.000
   ymin= -100.000    ymax=    6.000
numptsx=    41 numptsy=    54
deltaxcm= 200000.000 deltaycm= 200000.000
           ash mass integrated over isopach (g) = 0.1032E+16
                 ash mass extruded in event (g) = 0.1000E+16
          fuel mass integrated over isopach (g) = 0.1031E+09
                fuel mass extruded in event (g) = 0.1000E+09
```

62/69

SOFTWARE PROBLEM/CHANGE REPORT

01. SPCR Number: PA-SPCR-~~XXX~~ 117 [illegible]

02. Project Title: TSPA&I Code Development | 20-5708-762

03. SPCR Title: Change of The P(z,rho) Function Subroutine and the Integration Accuracy Parameters in the ASHPLUME code

04. AFFECTED SOFTWARE AND/OR DOCUMENTATION

1. ASHPLUME code Ashplume.f
2. ASHPLUME Version 1.0 - A code for Contaminated Ash Dispersal and Deposition; Technical Description and User's Guide

05. DESCRIPTION OF PROBLEM/CHANGE

The formulation for the probability density function for particle diffusion out of the eruption column as a function of height (z) and particle log-diameter (rho) as found in Suzuki (1983) becomes negative at heights approaching the top of the column. The formulation in Suzuki (1983) is as follows:

$$P(z,rho) = A \cdot Y(z,rho)\exp[-Y(z,rho)] \qquad (1)$$

where:

$A = \beta W_0/(V_0 H(1-(1+Y_0)\exp(-Y_0)))$
W_0 = The eruption velocity at height z=0 cm/s
V_0 = the particle terminal velocity at sea level (a function of rho) cm/s
H = The height of the eruption column Km
$Y(z,rho) = \beta(W(z)-V_0)/V_0$
β = A constant determining the fan shape of the column
W(z) = The particle velocity as a function of height = $W_0(1-z/H)$ cm/s
$Y_0 = Y(z=0,rho)$

The normalization constant A is found by integrating the probability density function over all heights and setting the intyegral equal to one. The problem is that using equation (1) and performing this procedure does not yield the value of A above. Also, P(z=h,rho) is negative which is fundamentally incorrect. The above formulation, however, does work if Y(z, rho) is equal to $\beta(W(z))/V_0$, I.e. P(z,rho) is universaly positive, and the definition of A above is correct.

Another problem identified is that the code requires a significant amount of runtime to complete it's calculations. The numerical integration routines have accuracy parameters associated with them. By choosing these parameters more judiciously than was done when the code was developed, improvements in runtime can be realized with little sacrifice of accuracy an the calculated values of ash and fuel depostion. For the gause-quadrature routine, this parameter is "n". For the romberg integration routines, this parameter is "eps".

REFERENCE

Suzuki, T. 1983. *A Theoretical Model for Dispersion of Tephra*. Terra Scientific Publishing Company. Tokyo, Japan.

63/69

06. PROBLEM SOLUTION/NEED FOR CHANGE

The solutions to the identified problems are as follows:

1. In the function subroutine P(z,rho), remove the V_0 term in the numerator in the definitions of both Y(z,rho) and Y_0.

2. Determine the minimum value of the parameter n in the gauss quadrature routine (page 34, line 26 of attachment A) and the maximum value of the eps parameters in the romberg integration routines (page 28, line 25 and page 30, line 25 of attachment A) that still provide reasonable accuracy.

07. Originator: Mark S. Jarzemba	Title: Research Engineer	Date: 5/8/97

PROJECT

08. Need by Date: 6/1/97	09. Approved: RGB	Disapproved:

10. Software Developer: Mark S. Jarzemba Mark Jarzemba	Date: 5/12/97
11. Element Manager: RGBaca	Date: 6/5/97

12. IMPLEMENTED SOLUTION

The solutions implemented to solve the above two identified problems are:

1. The function subroutine P(z,rho) on pages 16 and 17 of attachment A has been changed. The definition of variable yz on page 17, line 14 was commented out and replaced by the definition on line 15. The definition of variable y0 on page 17, line 16 was commented out and replaced by the definition on line 17.

2. The value of the parameter n on page 34, line 26 of attachment A was changed to n=20 and the values of the eps parameters on page 28, line 25 and page 30, line 25 of attachment A were changed to 1.d-4.

To verify that changes in the accuracy parameters have had no effect on calculated output. Two sets of runs were made; one with the old version of the code and one with the updated version. These are presented as attachments B and C respectively. The input file used to generate these realizations is found in attachment D.Some variation between the two sets is permissible because of change 1. Above. These differences are most noticeable for high wind speed realizations, such as realization 1 in the attachments.

13. Implemented By: Mark S. Jarzemba Mark Jarzemba	Date: 5/12/97

CNWRA Form TOP-5 (03/95)

Attachment A

64/69

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
c                                                                        c
      Program ashplume
c                                                                        c
c     This program calculates the ash and fuel areal mass density        c
c     at the x,y points on the earth's surface. The user is given        c
c     the option of recording particulate size distribution              c
c       information if so desired                                        c
c                                                                        c
c     Written by: Mark S. Jarzemba, 1/22/97 (this version)               c
c                                                                        c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c                                                                        c
c     maxd = maximum dimension of isopach points                         c
c     nmy  = maximum number of volcanoes                                 c
c                                                                        c
      parameter (maxd=500)
      parameter (nmy=1000)
      implicit double precision (a-h,o-z)
      implicit integer (i-n)
      character*60 version,title
      common /one/ beta,q,ashdenmin,ashdenmax,dmean,dsigma,fshape
      common /two/ h,werupt0,airden,airvis,c,u
      common /three/ fdmin,fdmax,fdmean,hmin,hmax,xmin,xmax
      common /threel/ dmin,dmax,rhomin,rhomax,rhomean
      common /four/ ymin,ymax,acutoff
      common /five/ x,y,udir,frhomin,frhomax,frhomean,drho,cmax
      common /six/ numptsx,numptsy
      common /seven/ Uran
      common /eight/ rhocut
      common /nine/ v,icount
      common /eleven/ ashrholow,ashrhohi
      common /twelve/ power,tdur
      common /thirteen/ numapts
      common /fifteen/ ipchar
      common /sixteen/ title
      dimension v(10000),xash(maxd,maxd),xfuel(maxd,maxd)
      open(unit=10,file='ashplume.out',status='unknown')
      version='ASHPLUME version 1.0'
c                                                                        c
c     Uran    = amount of waste extruded (1 container ~ 10MTU)           c
c     icount  = a counter for the string of random numbers               c
c             stored in v                                                c
c     iseed   = the random number seed                                   c
c     max     = the number of random numbers to get- 8 random nos.       c
c             per realization                                            c
c     numapts = the number of particle sizes to use in the histogram     c
c                 of particle size at the dose pt.                       c
c                                                                        c
      icount=1
      iseed=10
      numapts=30
      write (10,*)'iseed= ',iseed
      write (10,*) version
      max=10000
```

65/69

ASHPLUME CONSEQUENCE MODULE CODE SOFTWARE REQUIREMENTS DESCRIPTION

By
Mark S. Jarzemba

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

Reviewed by:

Randall D. Manteufel 12/10/96
Randall D. Manteufel

Approved by:

RG Baca 12/11/96
Robert G. Baca, Manager, Performance Assessment
CNWRA

ASHPLUME CONSEQUENCE MODULE CODE SOFTWARE REQUIREMENTS DESCRIPTION

1 INTRODUCTION

This Software Requirements Description (SRD) document is the first step in construction of the ASHPLUME code which is intended to be used in the Nuclear Regulatory Commission/Center for Nuclear Waste Regulatory Analyses (NRC/CNWRA) Iterative Performance Assessment (IPA). The ASHPLUME code is a stand-alone code that can also be used as a consequence module for the Total-system Performance Assessment (TPA) code. The TPA code simulates the performance of a geologic repository of nuclear High-Level Waste (HLW) at Yucca Mountain (YM), Nevada.

2 SOFTWARE FUNCTION

The ASHPLUME consequence module is being developed to determine the concentration of spent fuel (in grams of spent fuel per cm^2) at points on the surface of the earth after an extrusive volcanic event penetrates the repository and exhumes spent fuel. Using published data for wind velocity at the YM site and the pertinent volcanic parameters of events similar to those that could be expected at YM in the future, the ASHPLUME consequence module simulates the transport of contaminated particles (composed of spent fuel and ash) to surface points downwind. The spent fuel concentration will be used to calculate the dose to members of the local population after the event has deposited the contaminated isopachs.

3 TECHNICAL AND COMPUTATIONAL APPROACH

To assess the hypothetical radiation doses that would occur after a basaltic eruption, the distribution of radionuclides in the biosphere after such an event needs to be estimated. To estimate the distribution of radionuclides, it is assumed that the ash particles from the eruption are the carrier of the radionuclides. Models to predict the distribution of ash after an eruption have been developed with the intention of relating eruption magnitude to ash dispersion (Suzuki, 1983; Hopkins and Bridgeman, 1985; Glaze and Self, 1991). For ASHPLUME, the model described in Suzuki (1983) that relates eruption magnitude to ash distribution is modified to relate eruption magnitude to radionuclide distribution for YM based on a few simple assumptions. The model uses Monte Carlo sampling to determine the power and duration of the eruption, along with other properties of the ash particulates, and develops a radionuclide distribution from those sampled parameters. The radionuclide distribution can then be used to model dose to man.

A more complete discussion of the technical basis and computational approach for this module is described by Jarzemba (1996).

4 USER INTERFACE AND DATA FLOW

The code will be designed to accept input data for parameter values and distributions from a file and will write the results of the calculations to an output file.

4.1 INPUT TO ASHPLUME CODE

All of the input data for ASHPLUME is contained in the "ashplume.in" file. No other files/input will have an effect on the results of the calculation. The ashplume.in file contains the parameters necessary to describe the volcanic event scenario such as event duration, initial eruption velocity, mass of fuel to incorporate in the eruption, and wind speed and direction. Fuel and ash particle size information and parameter distribution information are also input.

4.2 OUTPUT FROM ASHPLUME CODE

Output data from ASHPLUME will include a listing of the parameters describing the volcanic eruption, the location and deposition of ash and fuel particulates, and particulate size information at selected dose points.

5 PROGRAMMING LANGUAGES

The ASHPLUME code is written in standard FORTRAN 77 as implemented in absoft FORTRAN for Macintosh. It is compliant with SUN SPARCompiler, Version 2.0 and Microsoft Fortran, version 5.0.

6 HARDWARE PLATFORMS

The ASHPLUME code will be developed for execution on Macintosh 8100/100, SUN machines using the UNIX operating systems, or DOS based personal computers.

7 GRAPHICS OUTPUT

No special graphics devices will be supported. Output will be in the form of ASCII files written in a format that can be read by spreadsheet programs, analysis software, and plotting packages.

8 PRE AND POST-PROCESSOR

The VOLCANO TPA consequence module may be used to generate the amount of waste to be distributed via this module. The results of the calculations will be used by the consequence module that performs the dose analyses in the TPA code. In NRC IPA phase 2 this module was DITTY. No post-processor is required or supported by the ASHPLUME code. The output files generated by the code will be designed so that they can be read easily by spreadsheet programs, analysis software, and plotting packages.

9 MATHEMATICAL MODEL, CONTROL FLOW, DATA FLOW, CONTROL LOGIC, AND DATA STRUCTURE

Please refer to the CNWRA report by Jarzemba (1996).

10 REFERENCES

Glaze L.S.and S. Self. 1991. *Ashfall Dispersal for the 16 September 1986 Eruption of Lascar, Chile, Calculated by a Turbulent Diffusion Model.* Geophysical Research Letters. **18**, pp. 1237-1240.

Hopkins, A.T. and C.J. Bridgeman. 1985. *A Volcanic Ash Transport Model and Analysis of Mount St. Helens Ashfall.* Journal of Geophysical Research. **90**, pp. 10,620-10,630.

Jarzemba, M.S. 1996. *Stochastic Radionuclide Distributions After a Basaltic Eruption for Performance Assessments of Yucca Mountain.* Journal article submitted to Nuclear Technology.

Suzuki, T. *A Theoretical Model for Dispersion of Tephra.* 1983. Arc Volcanism: Physics and Tectonics. Terra Scientific Publishing: Tokyo, Japan, pp. 95-113.

ASHPLUME Folder

[63] From: Randall Manteufel at CNWRA 12/6/96 4:16PM (1096 bytes: 1 ln) 69/69
To: Bruce Mabrito at CNWRA-OS2
cc: Mark Jarzemba, Amitava Ghosh at CNWRA-SUN, Robert Baca at CNWRA-OS2
Subject: SRD's
---------- Message Contents ----------

Bruce,

We have decided to develop stand-alone SRDs for the ASHPLUME & FAULTING computer programs given their 1) extensive use at the Center, 2) recognition by the NRC staff, and 3) anticipated future use in Center deliverables.

Earlier, we had planned to include descriptions of these 2 codes in the TPA SRD. But now we will issue individual SRDs.

You will receive 3 SRDs for the following codes:

1) FAULTING (lead: Amit Ghosh) by 12/15/96
2) ASHPLUME (lead: Mark Jarzemba) by 12/15/96
3) TPA 3.0 (lead: Randy Manteufel) by 1/15/97

Individual User Guides will also be completed for each of the 3 codes. Anticipated dates are:

1) FAULTING (lead: Amit Ghosh) by 1/20/97
2) ASHPLUME (lead: Mark Jarzemba) by 2/97 (?)
3) TPA 3.0 (lead: Randy Manteufel) by 3/17/97

Thanks.

-Randy