OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
ANALYSIS/MODEL COVER SHEET
Complete Only Applicable Items

1. QA: QA

2. ☒ Analysis — Check all that apply

Type of Analysis	☐ Engineering ☒ Performance Assessment ☐ Scientific
Intended Use of Analysis	☐ Input to Calculation ☒ Input to another Analysis or Model ☒ Input to Technical Document ☒ Input to other Technical Products

Describe use:
To calculate uncertainty distribution for infiltration rates.
Obtaining weighting factors for use by TSPA

3. ☐ Model — Check all that apply

Type of Model	☐ Conceptual Model ☐ Abstraction Model ☐ Mathematical Model ☐ System Model ☐ Process Model
Intended Use of Model	☐ Input to Calculation ☐ Input to another Model or Analysis ☐ Input to Technical Document ☐ Input to other Technical Products

Describe use:

4. Title:
Analysis of Infiltration Uncertainty

5. Document Identifier (including Rev. No. and Change No., if applicable):
ANL-NBS-HS-000027 REV 00

6. Total Attachments:
12

7. Attachment Numbers - No. of Pages in Each:
I-16, II-59, III-21, IV-73, V-5, VI-58, VII-22, VIII-35, IX-3, X-9, XI-29, XII-4

	Printed Name	Signature	Date
8. Originator	Ron McCurley	Ron McCurley	4/12/2000
9. Checker	Lawrence Sanchez	Lawrence C. Sanchez	4/12/2000
10. Lead/Supervisor	Ron McCurley	Ron McCurley	4/12/2000
11. Responsible Manager	Clifford K. Ho	Clifford K. Ho	4/12/00

12. Remarks:

Per Section 5.5.6 of AP-3.10Q, the responsible manager has determined that the subject AMR is not subject to AP-2.14Q review because the analysis does not affect a discipline or area other than the originating organization (Performance Assessment). The upstream supplier for this AMR was Joe Hevesi (USGS), and he worked closely with the originators in the development of the subject AMR to ensure that the inputs were used properly. The downstream user of the information resulting from this AMR is Performance Assessment (PA), which is also the originating organization of this work. PA leads, such as Mike Wilson, have worked with the originators during the development of this AMR. Nevertheless, both the upstream supplier and downstream customers were given the opportunity to provide informal comments on draft copies and to request a formal review if desired. However, it was determined that this analysis does not directly affect other organizations other than the originating organization. Therefore, no formal AP-2.14Q reviews were requested or determined to be necessary.

NM5507

DISCLAIMER

This contractor document was prepared for the U.S. Department of Energy (DOE), but has not undergone programmatic, policy, or publication review, and is provided for information only. The document provides preliminary information that may change based on new information or analysis, and represents a conservative treatment of parameters and assumptions to be used specifically for Total System Performance Assessment analyses. The document is a preliminary lower level contractor document and is not intended for publication or wide distribution.

Although this document has undergone technical reviews at the contractor organization, it has not undergone a DOE policy review. Therefore, the views and opinions of authors expressed may not state or reflect those of the DOE. However, in the interest of the rapid transfer of information, we are providing this document for your information per your request.

OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
ANALYSIS/MODEL COVER SHEET
Complete Only Applicable Items

1. QA: QA

2. ☒ Analysis — Check all that apply	3. ☐ Model — Check all that apply
Type of Analysis: ☐ Engineering; ☒ Performance Assessment; ☐ Scientific	Type of Model: ☐ Conceptual Model; ☐ Abstraction Model; ☐ Mathematical Model; ☐ System Model; ☐ Process Model
Intended Use of Analysis: ☐ Input to Calculation; ☒ Input to another Analysis or Model; ☒ Input to Technical Document; ☒ Input to other Technical Products	Intended Use of Model: ☐ Input to Calculation; ☐ Input to another Model or Analysis; ☐ Input to Technical Document; ☐ Input to other Technical Products
Describe use: To calculate uncertainty distribution for infiltration rates. Obtaining weighting factors for use by TSPA	Describe use:

4. Title:
Analysis of Infiltration Uncertainty

5. Document Identifier (including Rev. No. and Change No., if applicable):
ANL-NBS-HS-000027 REV 00

6. Total Attachments:
12

7. Attachment Numbers - No. of Pages in Each:
I-16, II-59, III-21, IV-73, V-5, VI-58, VII-22, VIII-35, IX-3, X-9, XI-29, XII-4

	Printed Name	Signature	Date
8. Originator	Ron McCurley	Ron McCurley	4/12/2000
9. Checker	Lawrence Sanchez	Lawrence C. Sanchez	4/12/2000
10. Lead/Supervisor	Ron McCurley	Ron McCurley	4/12/2000
11. Responsible Manager	Clifford K. Ho	Clifford K. Ho	4/12/00

12. Remarks:

Per Section 5.5.6 of AP-3.10Q, the responsible manager has determined that the subject AMR is not subject to AP-2.14Q review because the analysis does not affect a discipline or area other than the originating organization (Performance Assessment). The upstream supplier for this AMR was Joe Hevesi (USGS), and he worked closely with the originators in the development of the subject AMR to ensure that the inputs were used properly. The downstream user of the information resulting from this AMR is Performance Assessment (PA), which is also the originating organization of this work. PA leads, such as Mike Wilson, have worked with the originators during the development of this AMR. Nevertheless, both the upstream supplier and downstream customers were given the opportunity to provide informal comments on draft copies and to request a formal review if desired. However, it was determined that this analysis does not directly affect other organizations other than the originating organization. Therefore, no formal AP-2.14Q reviews were requested or determined to be necessary.

INFORMATION COPY
LAS VEGAS DOCUMENT CONTROL

OFFICE OF CIVILIAN RADIOACTIVE WASTE MANAGEMENT
ANALYSIS/MODEL REVISION RECORD

Complete Only Applicable Items

2. Analysis or Model Title:

Analysis of Infiltration Uncertainty

3. Document Identifier (including Rev. No. and Change No., if applicable):

ANL-NBS-HS-000027 REV 00

4. Revision/Change No.	5. Description of Revision/Change
REV 00	Initial Issue

CONTENTS

Page

1. PURPOSE 7

2. QUALITY ASSURANCE 9

3. COMPUTER SOFTWARE AND MODEL USAGE 9
   3.1 SOFTWARE Codes (UNCERTAINTY ANALYSIS) 9
   3.2 ROUTINES (UNCERTAINTY ANALYSIS) 10

4. INPUTS 13
   4.1 DATA AND PARAMETERS 13
      4.1.1 Uncertain Parameters 13
         4.1.1.1 Justification of Uncertainty Distributions 16
      4.1.2 Template Input Control File for INFIL 16
      4.1.3 Climate Data 16
      4.1.4 Watershed Data 17
      4.1.5 Sampling Control Parameters 18
      4.1.6 Control Input for Routines 18
   4.2 CRITERIA 19
   4.3 CODES AND STANDARDS 19

5. ASSUMPTIONS 19
   5.1 ASSUMPTIONS OF SPATIAL UNIFORMITY 20
   5.2 ASSUMPTION OF SimulatED DOMAIN EQUIVALENCE TO REPOSITORY FOOTPRINT 20
   5.3 ASSUMPTIONS USED FOR SAMPLING OF UNCERTAIN PARAMETERS FOR LHS 21
   5.4 ASSUMPTIONS USED FOR CALCULATION OF WEIGHTING FACTORS 21

6. ANALYSIS/MODEL 21
   6.1 LHS METHODOLOGY 22
      6.1.1 Overall Approach 22
      6.1.2 Methodology Applied to Glacial Transition Analysis 22
   6.2 RESULTS OF GLACIAL TRANSITION CLIMATE UNCERTAINTY ANALYSIS 23
   6.3 CALCULATION OF WEIGHTING FACTORS FOR TSPA 28
   6.4 SUMMARY OF UNCERTAINTY ANALYSIS 29

7. CONCLUSIONS 30
   7.1 SUMMARY 30
   7.2 RECOMMENDATIONS 30
   7.3 LIMITATION AND UNCERTAINTIES 31
      7.3.1 Epistemic Limitations and Uncertainties 31
      7.3.2 Aleatoric Limitations and Uncertainties 31
      7.3.3 Statistical Limitations and Uncertainties 31
   7.4 GENERAL IMPACT OF UNRESOLVED TECHNICAL INFORMATION 32

8. REFERENCES 33
8.1 DOCUMENTS CITED 33
8.2 CODES, STANDARDS, REGULATIONS, AND PROCEDURES 35
8.3 SOURCE DATA, LISTED BY DATA TRACKING NUMBER 36
8.4 SOFTWARE 36

ATTACHMENTS

ATTACHMENT I - Verification of INFIL V A_2.a1 as used on DEC Alpha at Sandia National Laboratories I-1

ATTACHMENT II - Listing of Source Code INFIL V A_2.a1 ..... II-1

ATTACHMENT III - Documentation of PREINFIL (to satisfy AP-SI.1Q) III-1

ATTACHMENT IV - Listing of Source Code PREINFIL(to satisfy AP-SI.1Q) IV-1

ATTACHMENT V - Documentation of POSTINFIL (to satisfy AP-SI.1Q) V-1

ATTACHMENT VI - Listing of Source Code POSTINFIL (to satisfy AP-SI.1Q) VI-1

ATTACHMENT VII - Listing of Sample Input for Routines VII-1

ATTACHMENT VIII - Sample Procedure Files VIII-1

ATTACHMENT IX - Flow Diagram of Uncertainty Analysis IX-1

ATTACHMENT X - Documentation for Other Routines Used in the Uncertainty Analysis (to Satisfy AP-SI.1Q) X-1

ATTACHMENT XI - Output File from LHS Containing Sampled Values of Input Parameters for all 100 Realizations XI-1

ATTACHMENT XII - Output Results from Analog Rate Maps Supplied by USGS Over Modeled Rectangular Region XII-1

FIGURES

Page

Figure 1-1. Modeled Region and Repository Footprint....8

Figure 3-1. Simplified Diagram of Process Flow for Uncertainty Analysis....13

Figure 6-1. Probability Distribution Function of Average Annual Infiltration for Glacial Transition Climate....26

Figure 6-2. Probability Distribution Function of Log Average Annual Infiltration for Glacial Transition Climate (weighting factors determined in Section 6.3)....27

Note: Additional figures are found in Attachment III

TABLES

Page

Table 3-1. Routines Used in Uncertainty Analysis of Infiltration Rates....11

Table 4-1. Uncertain Input Parameter Distributions for Glacial Transition Climate....14

Table 4-2. Description of Uncertain Input Parameters....15

Table 4-3. Climate Data for Uncertainty Analysis....16

Table 4-4. Watersheds Used in Glacial Climate Uncertainty Analysis....17

Table 6-1. Resulting Infiltration and $\log_{10}$ Infiltration Representative Rates for 100 Realizations....24

Table 6-2 Summary of Analog Infiltration Results and Weighting Factors Calculated in This Analysis....29

Table 7-1. Weights to be Used by TSPA for Choosing Number of Samples Which Use Low, Middle, or High Analog Infiltration Rate Maps....30

Note: Additional Tables are found in Attachments I, III, V, VII, VIII, X, XI, XII

ACRONYMS

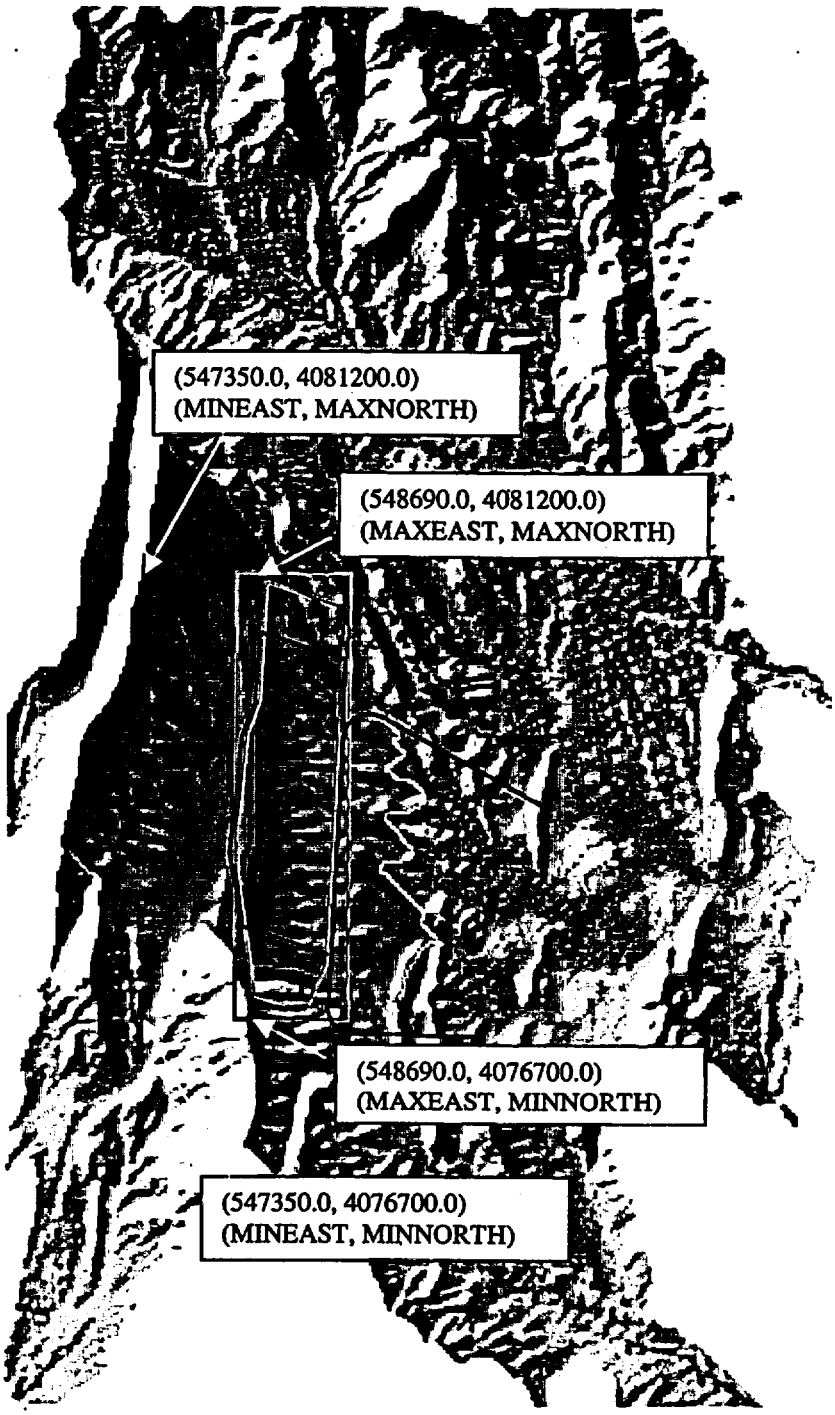
AMR	Analysis and Model Report
CDB	Computational Database
CFR	Code of Federal Regulations
CRWMS	Civilian Radioactive Waste Management Systems
DEC	Digital Equipment Corporation
DOE/RW	Department of Energy/Radioactive Waste
DTN	Data Tracking Number
DTR	Data Tracking Number
FE/FD	Finite Element/Finite Difference
LHS	Latin Hypercube Sampling
M&O	Management and Operating Contractor
NRC	U.S. Nuclear Regulatory Commission
PDF	Probability Density Function
QA	Quality Assurance
QARD	Quality Assurance Requirements and Description
RIP	Repository Integration Program
SAP	Software Activity Plan
SBR	Software Baseline Request
SR	Site Recommendation
SUR	Software User's Request
TBV	To Be Verified
TSPA	Total-System Performance Assessment
TSPA&I	Total-System Performance Assessment and Integration
URN	Unresolved Reference Number
USGS	U.S. Geological Survey
UZ	Unsaturated Zone
VA	Viability Assessment
VMS	Virtual Memory System (DEC operating system)
WIPP	Waste Isolation Pilot Plant

1. PURPOSE

The purpose of this analysis and model report (AMR) is to perform and document uncertainty analysis for the infiltration rates over the footprint of the repository. This uncertainty analysis was performed for the glacial transition climate. The results of this analysis yielded three weighting coefficients for infiltration (i.e., the probabilities for infiltration rates at the low, middle, and high bounds). The analysis is needed to address concerns raised by peer review groups, including the U.S. Nuclear Regulatory Commission (NRC), that the infiltration rate uncertainty was not adequately addressed in the Total-System Performance Assessment (TSPA)-Viability Assessment (VA).

TSPA-Site Recommendation (SR) has included three distinct climate regimes in the comprehensive repository performance analysis for Yucca Mountain; those being modern long term, monsoon, and glacial transition. For each of these climate regimes, infiltration rate maps have been provided to Performance Assessment for defining the hydrologic infiltration upper boundary condition for unsaturated zone flow and transport. For a more detailed discussion of these infiltration rate maps, see *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a). Each climate regime has three infiltration rate maps associated, including a lower and upper bound and a middle or medium value (equal to the average of the two boundary values). (Note that the middle rate map is not necessarily the mean or the median.) For each of these maps, which were obtained based on analog site climate data, a spatially averaged value was also calculated by the USGS. For this AMR, spatially averaged values were calculated for the lower, middle, and upper climate analogs only for the glacial transition climate regime using the modeled rectangular region in Figure 1-1. (For brevity, these maps will be referred to as the analog maps, and the corresponding averaged value as the analog value.) This was necessary to properly evaluate the location of each of these maps, using the corresponding analog value, within the appropriate uncertainty distribution calculated in this AMR. The primary objective of this uncertainty analysis is to use the uncertainty distribution and the results from selected analog site calculations done in *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a) to obtain weighting factors. These weighting factors are applied (in Repository Integration Program (RIP), Golder Associates, 1998) according to whether the analog maps produced for each climate regime, as provided by *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a) correspond to the lower, middle, and upper bound climate.

This analysis and model report provides an uncertainty analysis for the glacial transition climate only, since the climate model used in TSPA for the 10,000 year performance period is predominantly a glacial transition climate. The reader should note that this AMR provides an uncertainty analysis for only a single representative infiltration rate. This rate is obtained by spatially averaging the rates in a group of grid cells that range over the modeled rectangular region that encompasses the loaded footprint of the repository (see Figure 1-1). (The information that was contained in the infiltration rate map has been folded into this single averaged rate.) Details on how this averaged rate was arrived at are found in Sections 4.1.3 and 6.1.



NOTE: Outline of area (outer rectangular region) used in uncertainty analysis of infiltration relative to repository footprint regions (inner two)

UTM coordinates supplied by USGS (DTN: GS000308311221.004)

Figure 1-1. Modeled Region and Repository Footprint

This analysis incorporates the uncertainty inherent in probabilistic epistemic parameters (Rechard 1996, p. 4-3) by assigning, using known site characterization data, expert judgment and analog site data records, a range of values, and a distribution type to select infiltration model input parameters. These parameters were judged to account for the majority of uncertainty in the output infiltration rates. A detailed description of uncertainty input data and the justification for using the corresponding uncertain distributions is found in Sections 4.1.1 and 4.1.2.

The averaged infiltration rates calculated in this analysis used a set of uncertain input parameters (for the infiltration code INFIL) whose selection was constrained by schedule limitations. A future analysis might include some parameters not considered here, and exclude some that were considered.

As stated in the Work Direction and Planning Document, (CRWMS M&O 1999a) governing this activity, Paragraph 2 under the secondary heading "Infiltration" under "Scientific Approach and Primary Tasks," and in accordance with AP-3.10Q, *Analyses and Models*, this document evaluates the uncertainty in infiltration rate maps. Other subparts of this Work Direction and Planning Document (CRWMS M&O 1999a) are accomplished by other AMRs.

2. QUALITY ASSURANCE

This AMR is determined to be applicable to the QA program. In accordance with QAP-2-0, *Conduct of Activities*, an activity evaluation, *Conduct of Performance Assessment* (CRWMS M&O 1999b) has determined that the preparation of this technical document is subject to the requirements of the *Quality Assurance Requirements and Description* (QARD), DOE/RW-0333P (DOE 2000). Preparation of this AMR did not require the classification of items, in accordance with QAP-2-3, *Classification of Permanent Items*. The analyses conducted were not field activities. Therefore, a Determination of Importance Evaluation, in accordance with NLP-2-0, *Determination of Importance Evaluations,* was not required. The governing procedure for preparation of this AMR is AP-3.10Q, *Analysis and Models.*

3. COMPUTER SOFTWARE AND MODEL USAGE

The following two sections describe the software codes and the routines used in this analysis. All of the codes and routines used are in FORTRAN and compiled with the F77 compiler, with the exception of INFIL, which was compiled with the FORTRAN 90 compiler. All FORTRAN software was used on Platform DEC Alpha and Operating System Open VMS AXP V7 at Sandia National Laboratories.

3.1 SOFTWARE CODES (UNCERTAINTY ANALYSIS)

The analyses included the use of two major computer codes:

- INFIL (INFIL V.A_2.a1, compiled under FORTRAN90), a modified version (specifically to run on the platform described above in Section 3) of INFIL V. 2.0 performs climatic and hydrologic modeling to produce groundwater infiltration rate maps
- Latin Hypercube Sampling (LHS) (LHS V.2.50), a Monte Carlo engine using the LHS technique.

These two software codes require qualification under AP-SI.1Q, *Software Management*. INFIL V.A_2.a1 is a modified version of INFIL documented in DTN: GS960908312211.003. LHS is a software code that was qualified under the Waste Isolation Pilot Plant (WIPP) quality assurance (QA) requirements and a Software Activity Plan (SAP). A Software User's Request (SUR), a copy of software executable, along with a Software Baseline Request (SBR) have been submitted to the Software Configuration Secretariat to begin the qualification process for LHS (LHS V. 2.50) for the Yucca Mountain Site Characterization Project. The code INFIL (INFIL V.2.0 and INFIL V.A_2.a1) is currently in the process of being qualified according to AP-SI.1Q, *Software Management*, by the USGS. This code was developed and revised by the USGS for the purpose of modeling natural processes and features to produce output surface infiltration rate maps as described in *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a). It was also modified for this AMR to run on the DEC Alpha platform. It has been approved for use as unqualified software. A Software User's Request and Software Activity Plan have been submitted to the Software Configuration Secretariat to begin the qualification process for this code for the Yucca Mountain Site Characterization Project. This AMR is in no manner involved in maintenance or qualification of this software, except to verify that it produces essentially equivalent results on the platform used in this analysis as on the platform *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a). By "essentially equivalent results" means agreement to within 1 part in 10,000, a difference which is negligible relative to the precision or accuracy of the output infiltration rates developed by this software code. This is documented in Attachment I, containing a comparison of the infiltration rate, as found in the summary output file (of INFIL), of INFIL V A_2.a1 on DEC ALPHA at Sandia National Laboratories and of INFIL V 2.0 on a Windows NT 4.0 platform using a Pentium Pro PC at the USGS site. This documentation shows that the summary output net infiltration rates agree to about 1 part in 10 million (see Table I-1 in Attachment I). A listing of the source code can be found in Attachment II, Listing of Source Codes.

3.2 ROUTINES (UNCERTAINTY ANALYSIS)

Several FORTRAN routines (each consisting of multiple subroutines) were utilized for this analysis, including the development of a preprocessor, PREINFIL (PREINFIL V1.20), and postprocessor. POSTINFIL (POSTINFIL V2.50) for the code INFIL. The routines used were developed and used on the following platform; DEC Alpha running Open VMS AXP V7 (owned by the WIPP program) at Sandia National Laboratories.

The documentation to satisfy AP-SI.1Q, Paragraph 5.1.1.2, can be found in Attachments III and IV, for PREINFIL, and V and VI, for POSTINFIL.

The routines used in this analysis are listed in Table 3-1. All of the routines, with the exception of PREINFIL and POSTINFIL, were originally developed for the WIPP under the WIPP QA program. Documentation of the routines developed under WIPP is included in Attachment X of this AMR to begin the process of approval of the use of these routines for the Yucca Mountain Site Characterization Project. Use of the listed software routines, in this manner, is subject to the requirements listed in Section 5.1.1.2 of AP-SI.1Q. Documentation to satisfy the requirements of AP-SI.1Q is included in the Attachments as indicated in Table 3-1. Another routine used in this analysis, that was also developed under WIPP QA, is GROPECDB. However, this routine is

used for display of binary data only (data found in CDB files) and is exempt according to Section 2.1.5 of AP-SI.1Q.

The routines GENMESH (GENMESH V. 6.08) and MATSET (MATSET V. 9.01) set up a template computational database (CDB) file that will serve throughout the analysis as a dynamically changing reservoir of data from which the final stages of the uncertainty analysis and a potential sensitivity analysis can be accomplished. The CDB file also maintains a set of QA records that includes a complete history of what and when codes have written data to it.

The routine PRELHS (PRELHS V. 2.02) reads a file that contains the uncertain distribution data (DTN: GS000308311221.008) for the uncertain parameters and automatically creates the input file for LHS, the sampling engine. POSTLHS (POSTLHS V. 4.07) uses the output from LHS and an input user control file to create a new CDB file for each realization of sampled input data. (See Attachment VII, Listing of Sample Input Routines, and Attachment VIII, Sample Procedure Files, for examples of input control and procedure files and further elucidation of the interaction of these codes.)

Table 3-1. Routines Used in Uncertainty Analysis of Infiltration Rates

Code	Function	Attachment Where Documentation Resides
GENMESH* V.6.08	Generates FE/FD mesh or grid and original CDB file. Necessary for MATSET input.	X
MATSET* V.9.01	Sets up material property names in CDB file for sampling using LHS. Necessary for POSTLHS and multiple realization CDBs.	X
PRELHS* V.2.02	Creates input for LHS code using data file containing distribution parameters for uncertain parameters input to INFIL. Necessary for transfer of input distributions to create LHS input.	X
POSTLHS* V.4.07	Required transfers output of LHS consisting of sampled parameter values for multiple realizations into corresponding CDB file	X
PREINFIL V.1.20	Automatically transfers sampled values from CDB file from POSTLHS to INFIL input control files	III, IV
POSTINFIL V.2.50	Automatically transfers selected summary output from INFIL, including the spatial average infiltration rate for the calculated watershed into CDBs containing sampled parameter values.	V, VI
ALGEBRACDB* V.2.35	Required to calculate a mean infiltration rate for the repository footprint utilizing average infiltration rate and number of active cells information from all watersheds.	X

NOTES: An * (asterisk) after the name of a code indicates it was originally developed under QA requirements for the WIPP program. Each of the routines listed above utilize an input user control file (see Attachment VIII). Documentation to satisfy AP-SI.1Q is found in the attachments.

The above WIPP-QA-approved routines each have a User's Manual; citations for these manuals are included in the List of References. (Note that in some cases the version of the manual may not be the same as the version of the code because the version change in the code did not require a version change in the manual.)

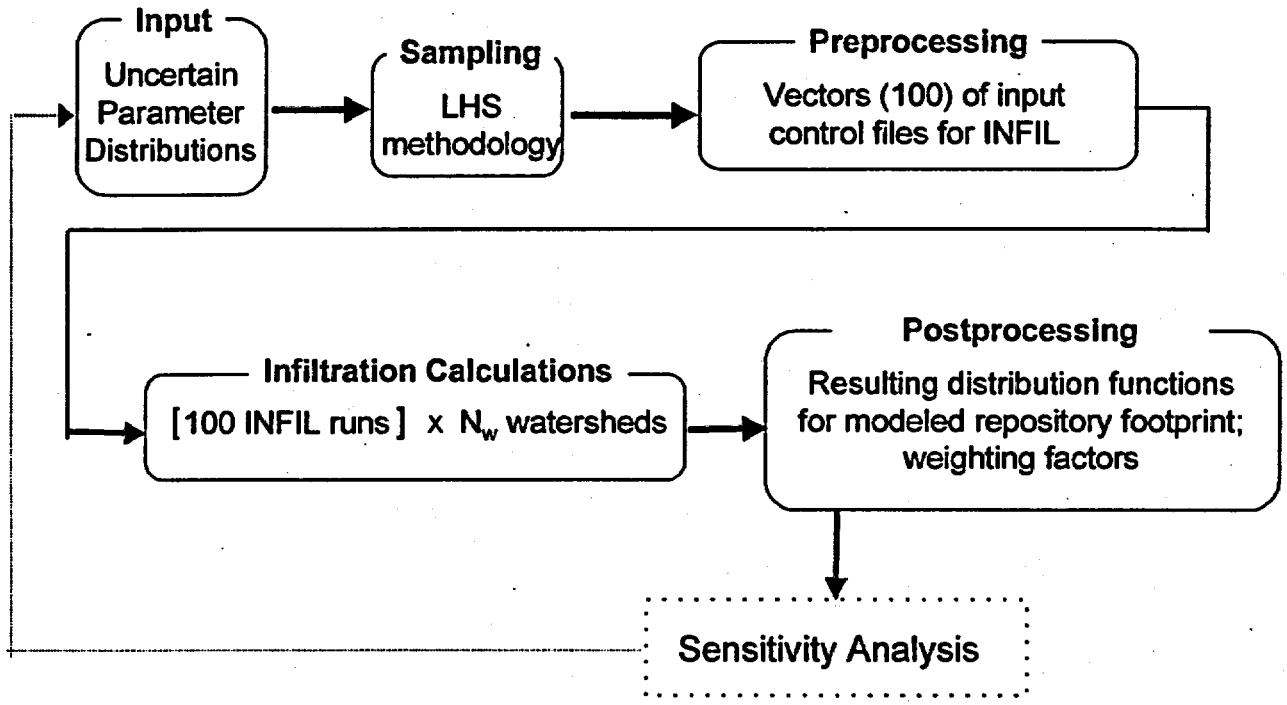
The routine PREINFIL was developed to automate the process of producing input control files formatted specifically for INFIL. This was essential for the process of readily producing numerous realizations of input for INFIL each containing a unique set of values for the sampled parameters. This enabled rapid and accurate transfer of data, in an automated environment, between LHS output and the input control file used by INFIL. The routine, POSTINFIL, was developed to automate the transfer of selected output, including the spatially averaged infiltration rate and the number of active cells for each watershed simulated, to intermediate CDB files. The CDB file is needed to exercise the uncertainty software for producing the uncertainty distributions. Both routines, PREINFIL and POSTINFIL, were developed and documented in this AMR.

Once the desired infiltration rate and watershed dimension data for each watershed included in the analysis has been stored in the CDB file, a routine called ALGEBRACDB (ALGEBRACDB V.2.35) computes the spatial average over the entire simulated domain (i.e., the rectangular region that includes the loaded repository footprint). The collection of these results for 100 realizations is displayed in a plot that shows the resulting infiltration rate distribution as a probability distribution function (PDF) (Figure 6-1, Figure 6-2). The data these plots represent can be found in Table 6-1.

Finally, for the uncertainty analysis, the predicted weights that can be applied to the analog maps are calculated (see Section 6.3).

Because so many routines are used in this analysis, a wiring diagram showing the sequence of code interactions is included in Attachment IX, *Flow Diagram of Uncertainty Analysis*. A simplified diagram is shown in Figure 3-1. Table 3-1 lists the routines used, a brief description of their function, and identifies the status. The use of these routines and input descriptions are described in the corresponding users manuals. (See Attachment VII, *Listing of Sample Input Routines*, and Attachment VIII, *Listing of Sample Procedure Files*, for examples of the input control and procedure files for further elucidation of the interaction of these codes.)

The documentation for the WIPP QA-approved codes (designated with an asterisk in Table 3-1) to satisfy AP-SI.1Q, Paragraph 5.1.1.2, can be found in Attachment X, *Documentation for Other Routines Used in the Uncertainty Analysis.*



Note: Dashed box indicates activities not performed for this analysis.

Figure 3-1. Simplified Diagram of Process Flow for Uncertainty Analysis

4. INPUTS

4.1 DATA AND PARAMETERS

The following five sections contain a brief summary of the input data and parameters used in the calculations for the analysis of the uncertainty in the infiltration rates. A sample listing of the input for each routine can be found in Attachment VII. A chart showing what input and output files are owned by each of the routines used is also included in Attachment VII.

4.1.1 Uncertain Parameters

The discussion in this section is limited to the (uncertain) analysis input (for INFIL) parameters (USGS 2000a, Section 6.10.2) that were judged as having potentially significant effects on the resultant infiltration rate uncertainty. The uncertainty analysis for the glacial transition climate regime includes a set of 12 (uncertain) parameters. These parameters, that were selected for sampling, are tabulated with distributions provided in Table 4-1 (see DTN in note below table). PRELHS, a routine (Section 3.2) uses the distribution data in this table as input in order to produce an input file for LHS. Table 4-2 provides a brief description of these parameters. The file containing the sampled parameter values per realization (for 100 realizations) resulting from LHS operating on these distributions is found in Attachment XI, Output Files from LHS Containing Sampled Values of Input Parameters

Table 4-1. Uncertain Input Parameter Distributions for Glacial Transition Climate

Idpram	Mean	Low Range	High Range	Distribution Type	Units
BRPERM	1.000	0.05	20.0	LOGNORMAL	NONE
BRPOROS	10.030	0.0000	0.040	NORMAL	NONE
BRZDEPTH	3.000	1.0000	5.000	NORMAL	METERS
ETCOEFFA	1.00	-5.000	-15.0	NORMAL	NONE
ETCOEFFB	1.040	0.540	1.540	NORMAL	NONE
FLAREA	100	0.01	0.490	NORMAL	NONE
POTETMUL	1.04.000	0.6000	1.400	NORMAL	NONE
PRECIPM	-10.000	0.6000	1.400	NORMAL	NONE
SNOPAR1	1.78	0.78	2.78	UNIFORM	NONE
SOILDEPM	3.000	0.5000	1.500	NORMAL	NONE
SOILPERM	1.000	0.05	20.0	LOGNORMAL	NONE
SUBPAR1	0.1	0	.2	UNIFORM	NONE

NOTE: Idpram (uncertain input parameter) values defined in Table 4-3.

For lognormal and normal distributions LHS assumes that the low and high values in the range are at the 1.0 and 99.0 percentile (Section 5.3).

The Data Tracking Number for this table is DTN: GS000308311221.008

Table 4-2. Description of Uncertain Input Parameters

Parameter	Description
BRPERM	Bedrock bulk saturated hydraulic conductivity multiplier
BRPOROS	Bedrock effective root-zone porosity
BRZDEPTH	Bedrock root-zone thickness multiplier
ETCOEFFA	First coefficient in expression for evapotranspiration
ETCOEFFB	Second coefficient in expression for evapotranspiration
FLAREA	Surface flow runoff area
POTETMUL	Daily evapotranspiration multiplier
PRECIPM	Daily precipitation multiplier
SNOPAR1	Snow-melt parameter
SOILDEPM	Soil zone thickness multiplier
SOILPERM	Soil saturated hydraulic conductivity multiplier
SUBPAR1	First term ("A1") in snow loss (sublimation) equation for temperature regime below freezing (i.e. $T_k \leq 0.0\ ^0C$)

NOTES: All 12 are used in the analyses for the infiltration rate uncertainty for the glacial transition climate.

For a more detailed description of input used in the INFIL code, refer to *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a).

4.1.1.1 Justification of Uncertainty Distributions

It is necessary to provide defensible explanation for the distribution types and ranges selected for the sampled input parameters used in this analysis (Table 4-1). All of the sampled input parameters were deemed to be normally distributed except for BRPERM, SOILPERM, SNOPAR1, and SUBPAR1; the first two were assigned lognormal distributions and the last two uniform distributions. See Section 6.10.2 in *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a) for details on the justification of the uncertainty distributions and their associated ranges.

4.1.2 Template Input Control File for INFIL

A template input user control file, for the software code INFIL V A_2.a1, was modified for each realization that was done, using the parameter values corresponding to each realization. The distributions and ranges of these parameters are described in Table 4-1. The values for these parameters for the 100 realizations can be found in Attachment XI, *Output File from LHS Containing Sampled Values of Input Parameters for All 100 Realizations.* The template input user control file was modified (by PREINFIL) by substitution of these sampled values to obtain the input user control file which was used for that realization. The template input user control file is found in Table VII-8, *Template Input User Control File for INFIL Version A_2.a1 Input to PREINFIL* (DTN: GS00308311221.011). The verification of the routine PREINFIL is found in Attachment III, *Documentation of PREINFIL (to satisfy AP-SI.1Q),* and Attachment IV, *Listing of Source Code PREINFIL(to satisfy AP-SI.1Q).* The documentation in Attachment III shows that PREINFIL correctly substitutes the sampled values from an arbitrary realization into the template input user control file.

4.1.3 Climate Data

Climate data consisting primarily of precipitation records were provided by USGS. Each climate regime has analog climate data associated with it; that analog data representing the average of the lower and upper climate bounds for that climate regime. The names and locations of the files on the DEC Alpha representing the glacial climate regime is given in Table 4-3.

Table 4-3. Climate Data for Uncertainty Analysis

Climate Analog	Data File Pathname on DEC Alpha	Status	Data Tracking Number
Glacial Transition	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.GLACIAL]TUKELAKE.INP	TBV	DTN GS000308311221.010

4.1.4 Watershed Data

The watersheds used to evaluate the spatial mean infiltration rate for the glacial transition climate are listed in Table 4-4.

In order to compute the spatial mean rate over the repository footprint for each watershed evaluated, the number of cells within the specified footprint boundary and the corresponding mean infiltration rate averaged over those cells were required. If the entire watershed falls within the footprint boundary, then the information corresponding to this watershed is output in a summary output file produced by INFIL. Otherwise, a subset of values corresponding to cells within the footprint boundary is averaged.

The coordinates of the loaded footprint boundary were provided by the U.S. Geological Survey (USGS) (Figure 1-1, DTN: GS000308311221.004). These coordinates (UTM) are determined from :

- Maximum East = 548690.0
- Maximum North = 4081200.0
- Minimum East = 547350.0
- Minimum North = 4076700.0.

Since the postprocessor to INFIL only considers rectangular regions, a rectangular approximation to the geometry of the loaded region (Figure 1-1) was provided by the USGS to determine the number of active cells within each watershed not falling completely inside this region. This is described more completely in Section 6.1.2.

Table 4-4. Watersheds Used in Glacial Climate Uncertainty Analysis

Name of Watershed	Symbolic Name	Data File Pathname on DEC Alpha	Status	Data Tracking Number (DTN:)
Wren Wash	ww1	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.GLACIAL.WW1]W1.W20	TBV	GS000308311221.004
Split Wash	sw1	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRA TION.INFIL.GLACIAL SW1]SW1.W20	TBV	GS000308311221.004
Coyote Wash	cw1	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRA TION.INFIL.GLACIAL.CW1]CW1.W20	TBV	GS000308311221.004
WT2 Wash	wt2	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRA TION.INFIL.GLACIAL.WT2]WT2.W20	TBV	GS000308311221.004
Drillhole Wash	dh3	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRA TION.INFIL.GLACIAL.DH3]DH2.W20	TBV	GS000308311221.004
Drillhole Wash	dh4	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRA TION.INFIL.GLACIAL.DH4]DH2.W.20	TBV	GS000308311221.004

Name of Watershed	Symbolic Name	Data File Pathname on DEC Alpha	Status	Data Tracking Number (DTN:)
Solitario Canyon Watersheds	us1	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.US1]US1.W20	TBV	GS000308311221.004
	se1	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE1]SE1.W20	TBV	GS000308311221.004
	se2	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE2]SE2.W20	TBV	GS000308311221.004
	se3	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE3]SE3.W20	TBV	GS000308311221.004
	se4	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE4]SE4.W20	TBV	GS000308311221.004
	se5	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE5]SE5.W20	TBV	GS000308311221.004
	se6	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE6]SE6.W20	TBV	GS000308311221.004
	se7	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE7]SE7.W20	TBV	GS000308311221.004
	se8	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE8]SE8.W20	TBV	GS000308311221.004
	se9	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE9]SE9.W20	TBV	GS000308311221.004
	se10	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTR A TION.INFIL.GLACIAL.SE10]SE10.W20	TBV	GS000308311221.004

4.1.5 Sampling Control Parameters

Two other parameters (one fairly incidental), specific to the sampling process, are the random seed used to initiate the Monte Carlo process (input to LHS) and the number of realizations (or samples), a control parameter. The random seeds used can be found in Attachment VII-7 that includes the PRELHS input control file that is used to produce the LHS input files used for each set of realizations. The number of realizations has already been stated to be 100. These are mathematical parameters and have no physical basis and, thus, no additional documentation is required.

4.1.6 Control Input for Routines

Each routine used in this analysis, tabulated in Section 3, has associated input and output files. Because of the linkage of codes used here, each of these routines, in addition to the two major codes used (LHS and INFIL), have output files that serve as input files to the code that is subsequently used in the analysis. Attachment IX, the Flow Diagram, showing the linkage of codes and routines, and Table VII-1 lists the files that serve to link and automate the entire procedure. The input files and their use for each routine used here, with the exceptions of the pre- and post- processor for the code INFIL, are described in their respective users manuals (Table 3-1). Sample input files and script (called also 'procedure') files, showing their use for each routine as used in these analyses, are included in Attachments VII and VIII. These inputs are control parameters, such as naming conventions, computational storage directives, etc., they have no basis in physical reality and, thus, no additional documentation is required.

4.2 CRITERIA

The NRC's *Issue Resolution Status Report Key Technical Issue: Total Performance Assessment and Integration* (TSPS&I) (NRC 1998) establishes generic technical acceptance criteria considered by the NRC staff to be essential to a defensible, transparent, and comprehensive assessment methodology for the repository system. These regulatory acceptance criteria address five fundamental elements of the U.S. Department of Energy TSPA model for the Yucca Mountain site:

1. Data and model justification, focusing on sufficiency of data to support the conceptual basis of the process model and abstractions

2. Data uncertainty and verification, focusing on technical basis for bounding assumptions and statistical representations of uncertainties and parameter variability

3. Model uncertainty, focusing on alternative conceptual models consistent with available site data

4. Model verification, focusing on testing of model abstractions using detailed process-level models and empirical observations

5. Integration, focusing on appropriate and consistent coupling of model abstractions.

Relevant to the topic of this AMR, elements 1 through 4 of the acceptance criteria are addressed herein and/or in the supporting calculation document(s). Element 5 of the NRC acceptance criteria, which strictly applies to the completed synthesis of process-level models and abstractions, will be addressed separately in the TSPA-SR.

This AMR was prepared to comply with the above NRC TSPA&I acceptance criteria, as well as the U.S. Department of Energy interim guidance (Dyer 1999) that requires the use of specified Subparts/Sections of the proposed NRC high-level waste rule, 10 CFR Part 63 (64 FR 8640). Subparts of this proposed rule that are particularly applicable to data include Subpart B, Section 15 (Site Characterization) and Subpart E, Section 114 (Requirements for Performance Assessment).

4.3 CODES AND STANDARDS

N/A. There are no codes and standards that apply to the analyses addressed in this AMR.

5. ASSUMPTIONS

The assumptions discussed below are summarized in Table 5-1.

Table 5-1. Assumptions

Assumption Description	Section Where Basis Is Described	Section Used In
Uniform variation in space of uncertain input parameters	5.1	4.1.1
Rectangular region analyzed adequate for developing weights for repository footprint	5.2	6.1.2
LHS assumes distribution endpoints represent 1.0 and 99.0 percentile	5.3	4.1.1, 6.1.1
Assumption of use of Glacial Transition climate analysis results for TSPA	5.4	6.2

5.1 ASSUMPTIONS OF SPATIAL UNIFORMITY

Because most of the parameters that were sampled are (linear) scaling factors for climatic, geological, and hydrological properties (e.g. soil/rock permeability, precipitation, etc.), the assumption is made, implicitly, that these properties will vary uniformly over the set of realizations independent of location. Thus, the original spatial heterogeneity (in a relative sense) of these properties is maintained. The basis for this assumption is that, by sampling on a set of scaling factors, the range of variability in the uncertainty of these climatic, geological and hydrological properties, that characterize the infiltration model, will be captured. For a complete discussion of these properties, see *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a, Section 6).

As discussed in Section 6.1.2 of this AMR, multiple watersheds were needed to produce a spatially averaged (over the footprint of the repository) infiltration rate. Each of these watersheds yields (for this analysis) a single associated (spatially averaged) infiltration rate and a weighted average is calculated for the simulated region. Each watershed has distinct geologic and hydrologic spatial variation (or heterogeneity). Since scaling factors for these properties are sampled in this analysis, this analysis (implicitly) assumes that all watersheds have the same relative variability in their associated climatic, geological and hydrological properties (thus, again, the original spatial heterogeneity is maintained). The basis for this assumption is essentially identical to that discussed in the preceding paragraph. Additional discussion regarding the selection of scaling factors for properties as sampled sampled parameters rather than the properties themselves is provided in *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a, Section 6.10.2).

This set of assumptions is utilized in selecting parameters to sample on, Section 4.1.1.

5.2 ASSUMPTION OF SIMULATED DOMAIN EQUIVALENCE TO REPOSITORY FOOTPRINT

Because the watersheds encompass areas extending outside the boundary of the simulated repository footprint region, it was necessary to compute a spatially averaged infiltration rate over

the subregion of each watershed that fell within the simulated domain. The simulated domain is, in actuality, the minimum area rectangular region that encloses the loaded footprint region. It is implicitly assumed that this is a reasonable approximation for computing infiltration rates representative of the loaded footprint region. Figure 1-1 shows this area outlined relative to the repository footprint. The basis for this assumption is that the behavior of infiltration in this region closely approximates the behavior of infiltration in the footprint (USGS 2000a, Section 6.10.1). For this analysis, the range of infiltration behavior is reasonably bounded and no further confirmation is required. Also the reader may refer to Table 6-2, note b.

This assumption is used in Section 6.1.2.

5.3 ASSUMPTIONS USED FOR SAMPLING OF UNCERTAIN PARAMETERS FOR LHS

For this AMR, zero pairwise correlation has been assumed between the sampled input parameters; that is, the parameters are assumed to vary independently of one another. The reason this zero correlation was assumed was that there was no evidence provided to suggest that the sampled input parameters were correlated.

For lognormal and normal distributions, LHS assumes that the low and high values in the range are at the 1.0 and 99.0 percentile. The basis for this assumption is that the 1.0 and 99.0 percentile values of a distribution contain nearly the entire range of the actual distribution. These values reasonably bound the distribution for this analysis and no further confirmation is required.

This group of assumptions is used both in the discussion of distributions for uncertain input parameters as listed in Section 4.1.1 and in the sampling process, as described in Section 6.1.1.

5.4 ASSUMPTIONS USED FOR CALCULATION OF WEIGHTING FACTORS

The glacial transition climate was chosen for estimation of the weighting factors, as derived for the lower, middle, and upper bound infiltration rate maps, because of the duration of this climate regime. The glacial-transition climate will dominate the majority of the simulated period (USGS 2000b), beginning at ~2000 years and lasting beyond the regulatory time period of 10,000 years. Consequently, no additional confirmation is required. It is used in Section 6.2.

6. ANALYSIS/MODEL

This is an analysis which uses a model, represented by the code INFIL (INFIL V A_2.a1) described in Section 3.1. No model was developed in this analysis. The use of INFIL in this analysis is both reasonable and necessary. This was the code used by TSPA for developing infiltration rate maps for use by the UZ models downstream. The primary rationale for this uncertainty analysis was to provide a basis for the numerical value of the weights chosen for the lower, middle, and upper analog infiltration rates.

6.1 LHS METHODOLOGY

6.1.1 Overall Approach

This analysis used a Monte Carlo technique, LHS, developed by McKay et al. (1979), and Iman and Shortencarier (1984). LHS is a form of stratified sampling, whereby, the sampling interval is divided into k intervals of equal probability, so that an equal number of samples, n, are drawn from each of the k intervals; thus, there will be nk samples in all.

The main advantage for using LHS is that greater confidence in the resulting output distribution, $f(\chi_1, \chi_2, \ldots, \chi_m)$ for uncertain parameters $\chi_1, \chi_2, \ldots, \chi_m$ can be obtained with far fewer samples than with Monte Carlo random sampling. A detailed discussion of the LHS methodology is beyond the scope of this AMR.

In order to avoid spurious pairwise correlation between uncorrelated input parameters, it is desirable to have at least 4/3 × m (m is the number of uncertain input parameters) samples (realizations) in the analysis. In this analysis, we chose to sample on 12 parameters, limited to 10 of these for the modern climate (listed in Section 4.1); thus, a minimum of 16 realizations would be necessary to do this. However, far more realizations (>>16) may be desirable for obtaining sufficient confidence in the quantification of the output distribution (i.e., infiltration rates). In this analysis 100 realizations (samples) were used. Limitations on the number of potential samples are driven by computational resources and the confidence in the input distributions and their ranges.

6.1.2 Methodology Applied to Glacial Transition Analysis

The long run times (CPU time) experienced in performing scoping calculations (~4.5 hours per realization for the larger watersheds along with a need to use a significantly long climate record for this future climate), motivated a decision to reduce the size of previously established watershed regions into multiple smaller watersheds that could be run in parallel. All of the watershed regions extended beyond the simulated rectangular region approximating the loaded footprint boundary. The USGS (DTN: GS000308311221.004) provided new watershed input files that significantly reduced the areas of the largest watershed region, by subdividing this region into smaller regions and eliminating much of the area from the simulated domain. Thus, for the glacial transition analysis, a total of 17 watersheds were used. The spatially averaged infiltration rate for each watershed is a spatial mean infiltration rate calculated as the averaged pointwise infiltration rates on the set of all nodal locations within the boundary of the designated rectangular subregion of the repository footprint. An areal (or number of active cells) weighted mean infiltration rate over the repository footprint was calculated using Equation 6-1. Using the definition of a weighted average for a discrete sum,

$$\vartheta_{rf} = \sum_{k=1}^{17} (\alpha_k \phi_k) / \sum_{k=1}^{17} \alpha_k \qquad \text{(Eq. 6-1)}$$

where

ϑ_{rf} = Mean (spatial) infiltration rate over the entire repository footprint

α_k = The weights are each represented by the area of corresponding watershed based on the number of active cells and the area of each cell; since the area of each cell is the same (900 m^2), this coefficient may also represent the number of active cells in watershed k

ϕ_k = The spatially averaged infiltration rate or hydrologic flux across the upper boundary of the unsaturated zone for each watershed, k

(The sum runs from 1 to 17 since there are 17 watersheds.)

Thus, ϑ_{rf} is calculated as a single number for each of the realizations done for this uncertainty analysis, and the calculated uncertainty distribution (the PDF) is a distribution for ϑ_{rf}.

All of the watersheds extend beyond this rectangular region's boundary. Thus, a subarea that corresponds to that part of the watershed within that boundary was used as the α_k. The average infiltration rate for each subpart of a watershed was calculated by averaging the infiltration rates over the cells within that subpart.

6.2 RESULTS OF GLACIAL TRANSITION CLIMATE UNCERTAINTY ANALYSIS

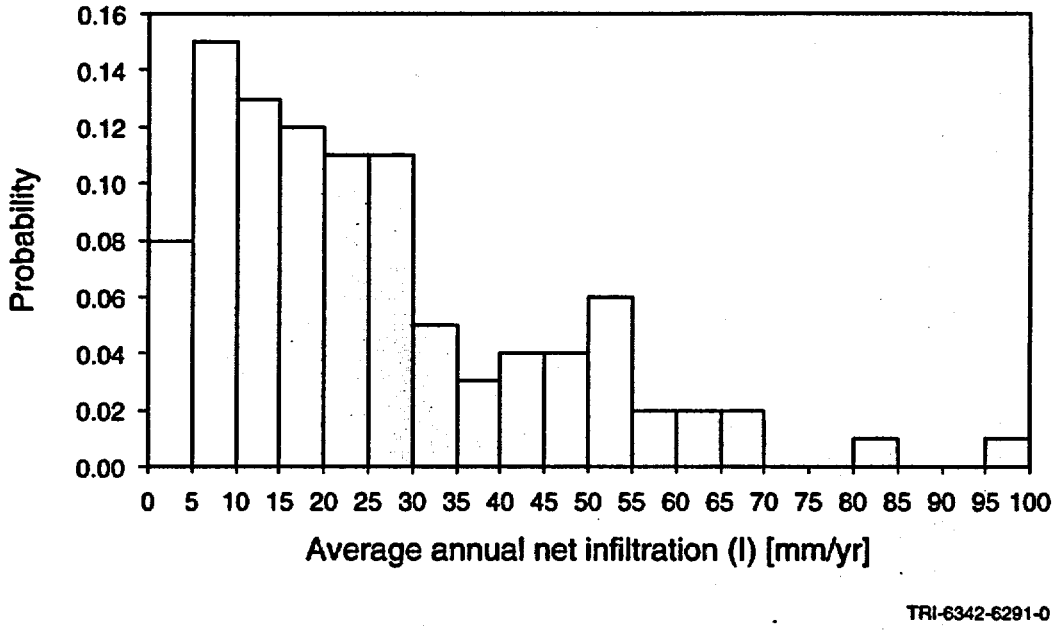
The USGS provided the output files (from INFIL V 2.0) (USGS 1999) containing infiltration rate maps that were produced from calculations corresponding to the lower, upper, and middle analog simulations as developed in *Simulation of Net Infiltration for Modern and Potential Future Climates* (USGS 2000a. Section 6.11.3). Each of these output files (containing infiltration rate map data) was used, in this analysis, for the calculation of a representative (analog) infiltration rate, averaged over an area identical to that used in these uncertainty calculations. Thus, three representative rates, corresponding to the lower, upper, and middle analog simulations were calculated for the simulated rectangular region. These rates are, here, called 'analog values'. The uncertainty distribution was used as a metric to determine where in the obtained distribution (i.e., the percentile) these representative analog values, corresponding to the mean, high, and low climate (for the glacial transition) would fall. A probability distribution function of these rates showing the locations of these analog values is shown in Figure 6-1. A probability distribution function of the logarithms (base 10) of the infiltration rates is shown in Figure 6-2. The approximate lognormal distribution of Figure 6-1 is predicted by the Central Limit Theorem. This figure is meant to display the data found in Table 6-1, along with the location in the distribution, of the analog values. The distribution displayed in Figure 6-1 for the uncertain infiltration rates (not the logs) has a mean of 25.5 [mm/yr] and a standard deviation of 19.2 [mm/yr] (see bottom of Table 6-1). The distribution displayed in Figure 6-2 for the uncertain log infiltration rates has a mean of 1.25, and a standard deviation of 0.42 (see bottom of Table 6-1).

Table 6-1. Resulting Infiltration and $\log_{10}$ Infiltration Representative Rates for 100 Realizations

R*	Infiltration Rate [mm/yr]	Log_{10} of Infiltration Rate
1	5.14E+01	1.711
2	2.33E+01	1.368
3	1.57E+01	1.197
4	2.25E+01	1.352
5	2.73E+01	1.436
6	1.64E+00	0.216
7	1.17E+01	1.066
8	2.48E+01	1.394
9	5.14E+01	1.711
10	3.07E+01	1.487
11	8.18E+01	1.913
12	2.07E+01	1.316
13	2.00E+01	1.301
14	1.50E+01	1.177
15	1.99E+01	1.299
16	4.95E+01	1.694
17	4.20E+01	1.623
18	9.35E+00	0.971
19	5.57E+01	1.746
20	3.65E+01	1.563
21	6.34E+00	0.802
22	6.80E+01	1.832
23	1.01E+01	1.003
24	1.14E+01	1.057
25	9.67E+01	1.986
26	8.60E+00	0.935
27	4.63E+01	1.666
28	2.66E+01	1.425
29	5.30E+01	1.724
30	2.23E+01	1.349
31	1.14E+01	1.058
32	6.18E+01	1.791
33	2.98E+01	1.475
34	5.77E+01	1.761
35	2.17E+01	1.336
36	5.17E+01	1.713
37	3.47E+00	0.540
38	2.28E+01	1.358
39	2.98E+01	1.475
40	3.43E+01	1.535
41	9.31E+00	0.969
42	1.05E+01	1.020
43	5.38E+01	1.731
44	2.97E+01	1.473
45	8.11E+00	0.909
46	4.39E+01	1.642
47	3.08E+01	1.488
48	8.29E+00	0.918
49	2.06E+01	1.313
50	5.45E+00	0.736
51	4.12E+01	1.614
52	3.64E+00	0.561
53	1.36E+01	1.133
54	8.76E+00	0.943
55	1.14E+01	1.058
56	6.40E+01	1.806
57	1.24E+01	1.094
58	3.47E+01	1.540
59	8.79E+00	0.944
60	1.56E+01	1.192
61	6.11E+00	0.786
62	1.31E+00	0.117
63	1.31E+01	1.118
64	1.94E+00	0.287
65	2.20E+01	1.341
66	1.55E+01	1.191
67	3.74E+01	1.573
68	1.26E+01	1.102
69	7.67E+00	0.885
70	2.66E+01	1.424
71	9.02E+00	0.955
72	1.23E+01	1.090
73	2.00E+01	1.302
74	3.29E+01	1.518
75	4.51E+01	1.654
76	1.15E+00	0.059
77	4.61E+01	1.663
78	1.96E+01	1.292
79	5.31E+01	1.725
80	6.72E+01	1.828
81	4.78E+00	0.680
82	6.00E+00	0.778
83	3.70E+01	1.568
84	2.33E+01	1.368

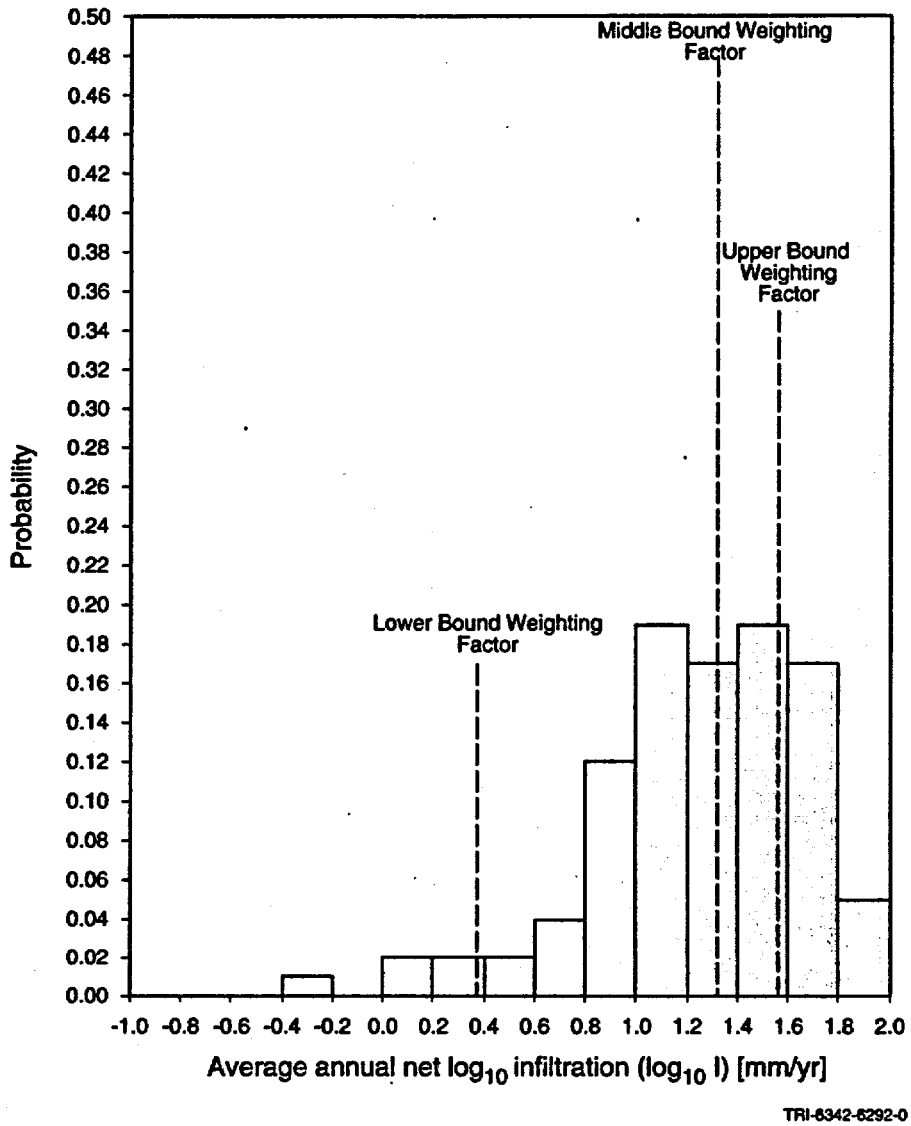
R*	Infiltration Rate [mm/yr]	Log_{10} of Infiltration Rate
85	2.89E+01	1.461
86	1.48E+01	1.169
87	5.04E-01	-0.298
88	1.58E+01	1.199
89	8.16E+00	0.912
90	1.17E+01	1.068
91	2.54E+01	1.405
92	1.70E+01	1.230
93	2.77E+01	1.443
94	1.61E+01	1.206
95	2.71E+01	1.434
96	1.65E+01	1.218
97	9.54E+00	0.980
98	1.53E+01	1.185
99	4.45E+01	1.648
100	2.55E+01	1.407
Mean=25.5		*Mean (log)=1.25*
StDev=19.2		***StDev (log)=0.42***

R* = Realization



NOTE: This plot only represents data presented in Table 6-1 (column 2).

Figure 6-1. Probability Distribution Function of Average Annual Infiltration for Glacial Transition Climate



NOTE: This plot only represents data presented in Table 6-1 (column 3).

Figure 6-2. Probability Distribution Function of Log Average Annual Infiltration for Glacial Transition Climate (weighting factors determined in Section 6.3)

6.3 CALCULATION OF WEIGHTING FACTORS FOR TSPA

The final desired product from these uncertainty analysis was to obtain defensible weighting factors, w_1, w_2, w_3, for the lower, medium, and upper analog climate infiltration rate maps (USGS 2000a, Section 6.11.3). Because the distribution of infiltration rates is nearly lognormal, it was decided that the logarithms of the infiltration rates would be used to evaluate the weighting coefficients. Thus, in log space, the following three equations hold;

$$w_1 + w_2 + w_3 = 1.0 \qquad \text{(Eq. 6-2)}$$

$$w_1 \bullet R_1 + w_2 \bullet R_2 + w_3 \bullet R_3 = \overline{R} \qquad \text{(Eq. 6-3)}$$

$$w_1 \bullet (R_1)^2 + w_2 \bullet (R_2)^2 + w_3 \bullet (R_3)^2 = \sigma^2 + \overline{R}^2 \qquad \text{(Eq. 6-4)}$$

where

w_1, w_2, and w_3 are the weights applied to the lower, middle, and upper analog climate infiltration rate maps; R_1, R_2, and R_3 are these same rate maps spatially averaged over the simulated rectangular region; σ and $\overline{R}$ are the standard deviation and the mean of the uncertainty distribution (all in log space); respectively. These equations are based on the definitions of normalized weighting factors, mean, and standard deviation. Equation 6-4 is merely a restatement of the definition of variance.

In log space, R_1=0.37, R_2=1.32, and R_3=1.59 (Table 6-2). And $\sigma = 0.42$ and $\overline{R} = 1.25$ (Table 6-1) correspond to the standard deviation and the mean of the uncertainty distribution in log space.

The weights must be the same, independent of climate regime (modern, glacial transition, monsoon) since this is the way these weights are treated in RIP. The results of the three independent uncertainty analyses performed for the three climate regimes do not produce equal weighting factors, nor should they be expected to. Since the RIP simulations will spend the majority of the simulated temporal domain in the glacial transition climate regime, the decision was made to use the weighting factors determined from the glacial transition climate uncertainty analysis. When Equations 6-2, 6-3, and 6-4 are solved to two significant decimal digits using the results from the glacial transition analysis, the values for w_1, w_2, and w_3 turn out to be 0.17, 0.48, and 0.35. Table 6-2 summarizes the results for the analog map representative values and the weighting factors corresponding to the lower, middle, and upper climates for the glacial transition climate regime.

Table 6-2 Summary of Analog Infiltration Results and Weighting Factors Calculated in This Analysis

Glacial Transition Climate	Analog Value[a,b]	Log_{10} Analog Value	Weighting Factor[c]
Low	2.36 [mm/yr]	0.37 [log_{10} mm/yr]	0.17
Middle	20.79 [mm/yr]	1.32 [log_{10} mm/yr]	0.48
High	39.22 [mm/yr]	1.59 [log_{10} mm/yr]	0.35

NOTE(s):

[a] The second column of numbers are calculated using the infiltration rate maps calculated by the USGS for the low, middle, and high glacial transition climates (DTN: GS000308311221.005) averaged over the simulated rectangular region (Figure 1-1). See Attachment XII. The routine POSTINFIL, documented in Section 3.2 and Attachment V, calculates the spatially averaged infiltration rate restricted to the simulated rectangular region

[b] The USGS-calculated analog values are 2.2, 19.8, and 37.3 mm/yr for the Low, Middle, and High Glacial Transition Climates, respectively (Table 7-13, USGS 2000a). The USGS calculated infiltration rates for the repository footprint, while the analysis documented in this report calculated infiltration rates for the outer rectangular region shown in Figure 1-1. The minor differences between the calculated means for the two results (result obtained for rectangular area and result obtained for potential repository area) are consistent with expected differences because of the slight differences in the two areas. The rectangular area is slightly larger than the potential repository area (see Figure 1-1) and includes locations with relatively higher estimates of net infiltration, which causes a slight increase in the means calculated for the rectangular area relative to the mean calculated for the potential repository area.

[c] Weighting factors are solutions to Equations 6-2, 6-3, and 6-4

6.4 SUMMARY OF UNCERTAINTY ANALYSIS

Uncertainty analysis was performed using 100 realizations of a selected set of 12 uncertain input parameters. This analysis results in a distribution of spatially averaged infiltration rates for a rectangular region closely approximating what is referred to as the loaded repository footprint. These distributions are displayed as probability distribution functions (Figures 6-1 and 6-2), and the locations relative to the uncertainty distribution (i.e., the percentile) of the corresponding lower, middle, and upper analog site results are shown in log space (vertical dashed lines in Figure 6-2). The height of these dashed lines represents the value of the weighting factors.

An inspection of Figure 6-2 shows that the middle analog value in log space is near, but above the mean value of the uncertainty distribution; an inspection of Figure 6-1 reveals that the value of the weight for the high analog value is reflected in the fact that the distribution is skewed toward the upper range of uncertainty.

7. CONCLUSIONS

7.1 SUMMARY

The uncertainty measure is provided by a probability distribution function (PDF) resulting from a set of 100 realizations (or vectors); each of which provides a unique representative infiltration rate. This representative rate is obtained by calculating the spatial average for the corresponding infiltration rate map, averaged over a rectangular region (see Figure 1-1) including the loaded footprint of the repository.

This analysis derived uncertainty distributions for the glacial transition climate regime, considered for TSPA-SR, since the time duration for this climate regime dominated the 10,000 year performance period. The uncertainty distributions, in conjunction with the analog site evaluations, are used to determine weighting factors for the analog maps that can be used by the TSPA. The results are based on input that is TBV. These results are displayed in Table 7-1.

Table 7-1. Weights to be Used by TSPA for Choosing Number of Samples Which Use Low, Middle, or High Analog Infiltration Rate Maps

Symbol	Weight	Climate
W_1	0.17	Low Analog
W_2	0.48	Middle Analog
W_3	0.35	High Analog

Weights to be used by TSPA for determining the number of samples that use the Low, Middle, or High Analog Infiltration Rate Maps

7.2 RECOMMENDATIONS

The following future work would enhance these analyses:

1. Determine a mean uncertainty distribution (in the form of a mean CCDF) of the mean infiltration rates by performing multiple sets of uncertainty analyses for a climate regime to provide an estimate of the measure of confidence in the obtained mean CCDF.

2. Perform a rigorous determination of uncertain input parameter correlation.

3. Implement an analysis to determine which parameters input to the infiltration code may significantly and adversely impact the criteria for compliance and licensing.

It may also be useful, in future work, to incorporate an uncertainty analysis into the process of selecting potential analog sites to represent the appropriate lower, upper, and middle climate for each climate regime.

To accomplish the first suggestion, it is necessary to perform multiple sets of realizations. For example, this analysis could be extended so that three sets of 100 realizations, using three distinct random seeds to begin the sampling process, which each produce an independent CCDF, from which a mean CCDF and a confidence measure of the mean CCDF, could be determined. The confidence measure is a stochastic estimate of the numerical robustness of the computer model used to obtain these distribution measures.

Accomplishing the second and third suggestions would require a sensitivity analysis, where the metrics for repository compliance (e.g., doses) are correlated with uncertain inputs to the infiltration model. Additional knowledge regarding the interrelationships between certain input parameters is necessary to define any significant correlation (pairwise) among these input parameters.

The final recommendation could be implemented by integration of the uncertainty/sensitivity analysis and results with selection of more appropriate analog sites for lower and upper climate bounds.

7.3 LIMITATION AND UNCERTAINTIES

The uncertainty analysis, by its very nature, incorporates uncertainties in the knowledge of parameter input values. The amount of uncertainty that is incorporated is based on multiple sources; that is, expert judgment, empirical data, model limitations, etc.

7.3.1 Epistemic Limitations and Uncertainties

The uncertainties in future climates and, thus, infiltration are directly related to the selection appropriate analog climate data; using this approach to forecast the future is problematic, given the chaotic nature of climate and the possibility that human activity may change the cyclical patterns. The climate records may be too short to adequately predict future precipitation patterns.

The effects of changes in the input parameter uncertainties can be qualitatively estimated from the results of the sensitivity analysis. Those uncertain input parameters which may show significant correlation with the output infiltration uncertainty will have significant impact, due to the nature of the propagation of uncertainty in the infiltration model, on the output infiltration rate distributions. The range in values of these significant parameters would be reflected in a qualitatively similar change in the output distribution range.

7.3.2 Aleatoric Limitations and Uncertainties

The glacial transition climate was the climate chosen for estimation of the weighting factors derived for the lower, middle, and upper bound climates used in RIP. The uncertainties in future climate onset and duration make this assumption problematic.

7.3.3 Statistical Limitations and Uncertainties

Limitations on the number of potential samples are driven by computational resources, runtime , and also by the number and size of watersheds used in the analysis. The limitations in

confidence in the input distributions and their ranges is propagated through the analysis to the output infiltration rate distributions and ranges, albeit somewhat nonlinearly.

7.4 GENERAL IMPACT OF UNRESOLVED TECHNICAL INFORMATION

This document may be affected by technical product input information that requires confirmation. Any changes to the document that may occur as a result of completing the confirmation activities will be reflected in subsequent revisions. The status of the input information quality may be confirmed by review of the Document Input System database.

8. REFERENCES

8.1 DOCUMENTS CITED

CRWMS M&O 1999a. *Climate and Infiltration Models for Use in PA (Rev. 01), ID: U1020, U1030; Activity: SPP5010, SPP5020.* Work Direction and Planning Document. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19990707.0101.

CRWMS M&O 1999b. *Conduct of Performance Assessment.* Activity Evaluation, September 30, 1999. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19991028.0092.

DOE (U.S. Department of Energy) 2000. *Quality Assurance Requirements and Description.* DOE/RW-0333P, Rev. 9. Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.19991028.0012.

Dyer, J.R. 1999. "Revised Interim Guidance Pending Issuance of New U.S. Nuclear Regulatory Commission (NRC) Regulations (Revision 01, July 22, 1999), for Yucca Mountain, Nevada." Letter from Dr. J.R. Dyer (DOE) to Dr. D.R. Wilkins (CRWMS M&O), September 3, 1999, OL&RC:SB-1714, with enclosure, "Interim Guidance Pending Issuance of New NRC Regulations for Yucca Mountain (Revision 01)." ACC: MOL.19990910.0079.

Gilkey, A.P. 1996. *WIPP PA User's Manual for ALGEBRACDB, Version 2.35* . WPO # 41864 Document Version 1.01. Albuquerque, New Mexico: NMERI. ACC: MOL.20000127.0064.

Golder Associates 1998. *Repository Integration Program RIP Integrated Probabilistic Simulator for Environmental Systems Theory Manual and User's Guide.* Redmond, Washington: Golder Associates. TIC: 238560.

Iman, R.L. and Shortencarier, M.J. 1984. *A FORTRAN 77 Program and User's Guide for the Generation of Latin Hypercube and Random Samples for Use with Computer Models.* NUREG/CR-3624. Albuquerque, New Mexico: Sandia National Laboratories. TIC: 224592.

McKay, M.D.; Beckman, R.J.; and Conover, W.J. 1979. "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code." *Technometrics, 21,* (2), 239-245. Alexandria, Virginia: American Statistical Association. TIC: 221741.

NRC (U.S. Nuclear Regulatory Commission) 1998. *Issue Resolution Status Report Key Technical Issue: Total System Performance Assessment and Integration.* Rev. 1. Washington, D.C.: U.S. Nuclear Regulatory Commission. ACC: MOL.19990105.0083.

Rath, J. 1996. *WIPP PA User's Manual for MATSET, Version 9.00*. WPO # 30688. Document Version 1.00. Albuquerque, New Mexico: NMERI. ACC: MOL.20000127.0048.

Rechard, R.P. 1996. *An Introduction to the Mechanics of Performance Assessment Using Examples of Calculations Done for the Waste Isolation Pilot Plant Between 1990 and 1992.* SAND93-1378 Revised. Albuquerque, New Mexico: Sandia National Laboratories. TIC: 245715.

Smith, L.N. 1995. *WIPP PA User's manual for POSTLHS.* WPO # 23551, Version Number: 4.06ZO. Albuquerque, New Mexico: Science Applications International Corporation. TIC: MOL.20000127.0065.

SNL (Sandia National Laboratories) 1995a. *WIPP PA User's Manual for GENMESH. Version 6.07.* WPO #23291. Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.200000310.0388.

SNL (Sandia National Laboratories) 1995b. *Requirements Document, Verification and Validation Plan for ALGEBRACDB, Version 2.35.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0109.

SNL (Sandia National Laboratories) 1995c. *Validation Document, ALGEBRACDB, Version 2.35.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0110.

SNL (Sandia National Laboratories) 1995d. *Validation Document, GENMESH, Version 6.07ZO.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0117.

SNL (Sandia National Laboratories) 1995e. *Requirements Document, Verification and Validation Plan, POSTLHS, Version 4.06ZO.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0123

SNL (Sandia National Laboratories) 1995f. *Validation Document, POSTLHS, Version 4.06ZO.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0124.

SNL (Sandia National Laboratories) 1996a. *WIPP PA User's Manual for LHS Version 2.41.* Sandia WIPP Central Files ERMS#230732. Albuquerque, New Mexico: Sandia National Laboratories. URN-0116. Submit to RPC.

SNL (Sandia National Laboratories) 1996b. *WIPP PA User's Manual for GROPECDB (Version 2.12). Document Version 1.0.* WPO # 37496. Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000310.0381

SNL (Sandia National Laboratories) 1996c. *Requirements Document, Verification and Validation Plan, GENMESH, Version 6.08.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0118.

SNL (Sandia National Laboratories) 1996d. *Requirements Document, Verification and Validation Plan, MATSET, Version 9.00.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0122.

SNL (Sandia National Laboratories) 1996e. *Validation Document, MATSET, Version 9.00.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0121.

SNL (Sandia National Laboratories) 1997a. *Requirements Document, Verification and Validation Plan, PRELHS, Version 2.20.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0125.

SNL (Sandia National Laboratories) 1997b. *Validation Document Reviewer's Form, PRELHS, Version 2.20.* Albuquerque, New Mexico: Sandia National Laboratories. ACC: MOL.20000308.0126.

USGS (U.S. Geological Survey) 2000a. *Simulation of Net Infiltration for Modern and Potential Future Climate.* Input Transmittal 00132T. Denver, Colorado: US Geological Survey. ACC: MOL.20000330.0421.

USGS (U.S. Geological Survey) 2000b. *Future Climate Analysis.* ANL-NBS-GS-000008 REV 00. Denver, Colorado: US Geological Survey. Submit to RPC URN-0004.

8.2 CODES, STANDARDS, REGULATIONS, AND PROCEDURES

64 FR 8640. Disposal of High-Level Radioactive Wastes in a Proposed Geologic Repository at Yucca Mountain, Nevada. Proposed rule 10 CFR 63. Readily Available

AP-3.10Q, Rev. 2, ICN 0. *Analyses and Models.* Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.20000217.0246.

AP-SI.1Q, Rev. 2, ICN 4. *Software Management.* Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.20000223.0508.

NLP-2-0. Rev. 5. *Determination of Importance Evaluations.* Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19981116.0120.

QAP-2-0, Rev. 5, ICN 1. *Conduct of Activities.* Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19991109.0221.

QAP-2-3, Rev. 10. *Classification of Permanent Items.* Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19990316.0006.

8.3 SOURCE DATA, LISTED BY DATA TRACKING NUMBER

GS000308311221.004. Preliminary Geospatial Input Data for INFIL V 2.0 FY1999. Submittal date: 03/01/2000.

GS000308311221.005: Preliminary Net Infiltration Modeling Results for 3 Climate Scenarios for FY99. Submittal date: 03/01/2000.

GS000308311221.008. Preliminary Estimates of Input Parameter Distributions Used for Infiltration Uncertainty Analysis. Submittal date: 03/13/2000.

GS000308311221.010. Preliminary Developed Daily Climate Data from Tule Lake, California, Used for Infiltration Uncertainty Analysis. Submittal date: 03/07/2000.

GS000308311221.011: Template Files for Uncertainty Analyses. Submittal date: 03/13/2000.

8.4 SOFTWARE

Sandia National Laboratory. *Software Code: LHS V2.50*. 1996. V2.50. 110205-2.50-00.

Sandia National Laboratory. *Software Code: INFIL V.A_2.a1*. 1999. VA_2.a1. 10253-A2.a1-00.

U.S. Geological Survey. *Software Code: INFIL V2.0*. 2000. V2.0. 10127-2.0-00.

GS960908312211.003. Conceptual and Numerical Model of Infiltration at Yucca Mountain, Nevada. Submittal date: 09/12/1996.

ATTACHMENT I

Verification of INFIL V A_2.a1 as Used on DEC Alpha at SANDIA NATIONAL LABORATORIES

This attachment contains a comparison between Summary Output results from execution of the code INFIL V A_2.a1 on the DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories and the code INFIL V 2.0 using Windows NT 4.0 on a Pentium Pro PC by the USGS. Table I-1 shows the difference in the results to be negligible. Both codes (INFIL V A_2.a1 and INFIL V 2.0) are being qualified by the USGS.

Comparison of results as displayed in the summary output file using equivalent input (see Table I-1) for platform used at Sandia National Laboratories (Section 3.1) with corresponding results from the version of the code as executed by the USGS on a system designated by:

User Location: U.S. Geological Survey, Room 5000E, Placer Hall, 6000 J Street, Sacramento, California 95819-6129
Computer Identification: SM321276 with a USGS specific hostname P720dcasr
Platform identification information: Pentium Pro PC
Operating System: Windows NT 4.0

Table I-1. Comparison of Summary Output Results from version of INFIL at Sandia National Laboratories with those from INFIL as used by USGS

Ouput Parameter	Version		Relative Error
	SNL	USGS	
Precipitation	290.532770	290.532762	2.75356E-08
Rain	233.901757	233.901750	2.99271E-08
Snowfall	56.631013	56.631012	1.76582E-08
Snow-cover	631.284972	631.285751	1.23399E-06
Snow-melt	50.640619	50.640612	1.38229E-07
Sublimation	5.261948	5.261954	1.14026E-06
Potential Evapotranspiration	832.586015	832.585923	1.10499E-07
ACTUAL EVAPOTRANSPIRATION	262.509388	262.509371	6.47596E-08
Change in Soil Moisture	0.727825	0.727825	0
*Net Infiltration**	*21.305163*	*21.305166*	*1.40811E-07*
Runoff Generation	0.690760	0.690760	0
Cumulative Daily Run-on	0.094091	0.094091	0

NOTE: Net Infiltration is the output parameter of importance.

Table I-2. User Control File Used to for Verification of INFIL (version A_2.a1) on Platform DEC Alpha and Operating System Open VMS AXP V7 at Sandia National Laboratories

```
INFIL2a1.ctl: Solitario East, Tule Lake, se10-2a1-gm1-k2-w20 (11/18/1999)
1   0.00000001                                  IROUT(1 = coupled, 0 = uncoupled, -1 = flow routing off, -2 = infil off), IFRTOL
2   2     1.78                                      ISNOW, ISNWMOD, SNOPAR1
3   0.1   0.3                                   ISUBLIM, SUBPAR1, SUBPAR2
0   1.                                          IPPTTEST (1 for testing), PPTTEST = constant
0   .0                                          IETTEST (1 for testing), ETTEST = constant
30.0                                                CELSIZE (node spacing (meters): using for flow volume calculations)
544721.0  4072203.0                             xcfs,ycfs: coords for discharge cell
1950  1    1996    366   1   5                 yr1 = start year, dnn1 = start day, yr2 = end year, dn2 = end day
1     1    1    1                              multipliers ( pptfact, etfact, imbfact, sksfact)
1.0   0    0.4                                 SDFACT (soil depth multiplier), IVEGC (set to 1 for map data), FVEGC (use if IVEGC =
0)
1.0   0.8   0.8    0.2                          ROOTF1,ROOTF2,ROOTF3,ROOTF4
1.0   1.0   0.5    0.1                          MAXWGT1, MAXWGT2, MAXWGT3, MAXWGT4
0.3   2.0   3.0    2.0                           RDEPTH1,RDEPTH2,RDEPTH3,RDEPTH4,RDEPTHF
0.02  1.0  .25                                  RKPOR, RKMMFACT, FLAREA
1   1   0                                       INFMOD, ETMOD, RUNMOD
 -10.0  1.04                                 BARSOIL1, BARSOIL2:  bare soil et parameters
3   17.3  11.74                                 IAIRTEMP = 1 for new air temp model, ATEMP1 = avg. air temp, ATEMP2 = air temp
seasonal deviation
2                                               HSTEP: time step for PET model (hours)
5  181  3                                       PPTYUC (=5 diminished elev. correlation, =2 for 4JA , =1 for simple elevation
transfer), AAPREPX, IPPTDAT
Tulelake.inp                                    input file name:  daily precip
     0      1    0.5      1     1               dpthflag, irtz, delvwcf, moistcr, fracmod
     0      1.75                                 IVWCFLG,   vwcfact
se10.w20                                         input file name:  map parameters (*.inp)
-1
se10-gm1.2a1                                       dayall: average daily mass balance terms
1    1                                        ndaymap, imap
70   1995
se10-gm1.2a2
se10-gm1.2a3                                       output file name:  daily mass balance
se10-gm1.2a4                                       output file name:  total daily fluxes
1      1                                        dbgflag, dbgflag2
se10-gm1.2a5                                       output file name:  annual mass balance terms
0                                               IDEBUG:  debugging option parameter
se10-gm1.2a6                                       new debug file
se10-gm1                                         map output:  annual totals or mult-year averages

--
--------------- Parameters for dynamic root-zone function -------------------------------------
--     depth  rtza   rtzb   rtzc  rtzd  bsoil    delvwc
--     m
     4
     1    0.5     15      3     3     2     0.30      0.50
     2    1.5     15      4     2     2     0.20      0.50
```

```
    3    4.5     10    1.5     1     2     0.10      0.50
    4    6.0     10    1.5     1     2     0.05      0.25
--
--------------- Soil Properties (Brooks & Corey/van Genuchten combined) ----------------------------------------
--     fdcp   etresidpor    beta  alpah ksat     PE        B     n     vg-alpha sorp     SOILP   potis
--                                        Kg s /m^3J/Kg                 1/(J/Kg) ???              J/Kg
   10
    1  0.242  0.054  0.366  -3.5  1.26  5.6E-04 -1.19E+00  4.72  1.24  5.2E-01    0.390    0.05 -1.0E+02
    2  0.173  0.023  0.315  -3.5  1.26  1.2E-03 -9.41E-01  3.70  1.31  6.2E-01    0.500    0.05 -1.0E+02
    3  0.163  0.017  0.325  -3.5  1.26  1.3E-03 -8.60E-01  3.36  1.36  6.6E-01    0.510    0.05 -1.0E+02
    4  0.073  0.002  0.281  -3.5  1.26  3.8E-03 -6.22E-01  2.18  1.62  8.7E-01    0.700    0.05 -1.0E+02
    5  0.200  0.035  0.330  -3.5  1.26  6.7E-04 -1.07E+00  4.14  1.78  5.6E-01    0.400    0.05 -1.0E+02
    6  0.150  0.011  0.339  -3.5  1.26  2.7E-03 -7.55E-01  3.06  1.40  7.4E-01    0.700    0.05 -1.0E+02
    7  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
    8  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
    9  0.189  0.028  0.322  -3.5  1.26  5.7E-04 -1.08E+00  3.88  1.30  5.5E-01    0.370    0.05 -1.0E+02
   10  0.189  0.028  0.322  -3.5  1.26  5.7E-05 -1.08E+00  3.88  1.00  5.5E-01    0.037    0.05 -1.0E+02
--

---------------Rockl Properties (Brooks & Corey/van Genuchten combined)----------------------------------------
--     rkcp   etresidpor    beta  alpha ksat     PE        B     n     vg-alpha fracks   imbibe  potir
--                                        Kg s /m^3J/Kg                 1/(J/Kg) Kg s /m^3mm/dy   J/Kg
  129
    1    0.35    0.065    0.366    -1.5    1.26   0.0011    -0.93     3.6    1.25     0.52  0.00056     500    -100
    2   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25      890  3.7E-07    0.41    -100
    3   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23    42000 0.000052   46.66    -100
    4   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.09    -100
    5   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23    42000 0.000052   46.66    -100
    6   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25      890  3.7E-07    0.41    -100
    7   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23    42000 0.000052   46.66    -100
    8   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.09    -100
    9   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23    42000 0.000052   46.66    -100
   10   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.09    -100
   11   0.105     0.02     0.11    -1.5    1.26  4.0E-09       -6    3.92    1.47       64  2.4E-07    0.06    -100
   12   0.388   -0.039    0.393    -1.5    1.26 0.000053     -242    4.57    1.38     2400 0.000053    2.74    -100
   13    0.43    0.044    0.435    -1.5    1.26   0.0004    -1790    6.81    1.23    18000   0.0004   13.83    -100
   14   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.09    -100
   15   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23    42000 0.000052   46.66    -100
   16   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.09    -100
   17   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25      890  3.7E-07    0.35    -100
   18   0.248     0.01    0.253    -1.5    1.26  3.8E-06    -82.3    2.71    1.84      830  3.9E-06    3.34    -100
   19   0.159     0.01    0.164    -1.5    1.26  1.2E-06     -140    3.61    1.53     1400  1.3E-06    1.13    -100
   20   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   21   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   22   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   23   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   24   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   25   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   26   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   27   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
   28   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69      120  1.4E-07    0.06    -100
```

29	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
30	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
31	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
32	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
33	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
34	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
35	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
36	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
37	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
38	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
39	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
40	0.494	0.05	0.499	-1.5	1.26	0.000088	-341	4.19	1.43	3400	0.000088	75.62	-100
41	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
42	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
43	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
44	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-26.9	5.6	1.29	270	2.1E-07	0.09	-100
45	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
99	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
46	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
47	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
48	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
49	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
50	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
51	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
52	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
53	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
54	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
55	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
56	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
57	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
58	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
59	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
60	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
61	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
62	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
63	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
64	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
65	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
66	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
67	0.031	0.018	0.036	-1.5	1.26	5.0E-09	-0.6	3.39	1.58	10	2.4E-07	0.17	-100
68	0.168	0.021	0.173	-1.5	1.26	7.3E-08	-125	5.36	1.31	1300	1.3E-07	0.07	-100
69	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.02	-100
70	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
71	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.01	-100
72	0.317	0.023	0.322	-1.5	1.26	3.8E-08	-181	4.01	1.46	1800	9.9E-08	1.65	-100
73	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
74	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
75	0.281	0.051	0.286	-1.5	1.26	1.7E-08	-17.5	4.01	1.45	180	7.8E-08	0.02	-100
76	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
77	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
201	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100

202	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.06	-100
203	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
204	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	120	1.4E-07	500	-100
205	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
206	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
207	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
208	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
209	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
210	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
211	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
212	0	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
213	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
214	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
301	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
302	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
303	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
304	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
305	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
306	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
307	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
308	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
309	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
310	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
311	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
312	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
313	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
314	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
315	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
316	0.159	0.01	0.164	-1.5	1.26	1.2E-06	-140	3.61	1.53	1400	1.3E-06	1.13	-100
317	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
318	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
319	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
320	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
321	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
322	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
323	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
324	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
325	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
326	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
327	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
328	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
329	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
330	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
331	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
332	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
333	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
334	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
335	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
336	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
345	0.097	0.01	0.102	-1.5	1.26	1.3E-07	-154.6	5.02	1.56	1560	2.9E-07	0.24	-100

Table I-3. Summary Output Produced for Verification of INFIL (version A_2.a1) on Platform DEC Alpha running Open VMS AXP V7 at Sandia National Laboratories

```
INFIL2a1.ctl: Solitario East, Tule Lake, se10-2a1-gm1-k2-w20 (11/18/1999)
Daily precipitation input:              Tulelake.inp
Watershed modeling domain parameters:   se10.w20
Average daily mass balance terms:       se10-gm1.2a1
24-hour mass balance map:               se10-gm1.2a2
Annual mass balance map:                se10-gm1.1
Average annual mass balance map:        se10-gm1.2a4
Summary statistics output:              se10-gm1.2a3
Debugging output:                       se10-gm1.2a5


Total number of days read in   =    12053
Total daily precip             =   9660.7
Average annual precip (mm)     =    292.8
Maximum daily precip (mm)      =     51.3


TOTAL NUMBER OF LOCATIONS      =       69
AVERAGE ELEVATION OF SAMPLE    =   1339.3
MAXIMUM ELEVATION OF SAMPLE    =   1377.0
MINIMUM ELEVATION OF SAMPLE    =   1301.0
AVERAGE SOIL DEPTH (M)         =    2.124
AVERAGE SLOPE OF SAMPLE        =     12.8
MAXIMUM SLOPE OF SAMPLE        =     23.0
NUMBER OF ACTIVE LOCATIONS     =       17


  Yr  Dy    Precip      Rain      Snow  Sn-cover  Snowmelt    Sublim       PET     PETRS  Evapotrs  Del-soil   Net-inf    Runoff  Out-
flow    Mass-balance
1950 365   288.823   218.232    70.591   432.356    60.488     5.658   835.723   663.272   242.653    36.067     0.000     0.000
0.000  -0.896813276E-10
1951 365   212.451   120.978    91.473   736.623    54.355     7.819   842.857   668.934   194.543   -19.922     0.711     0.000
0.000  -0.670113706E-09
1952 366   271.664   232.269    39.395   354.124    64.979     4.581   812.532   644.867   292.275    -8.315    13.289     0.000
0.000  -0.118735058E-08
1953 365   367.872   313.935    53.937   178.650    51.584     5.932   799.670   634.659   315.938    37.982    11.599     0.000
0.000  -0.161001923E-08
1954 365   212.451   135.956    76.495   495.140    53.006     7.244   813.631   645.739   223.574   -55.092    20.480     0.000
0.000  -0.254417724E-08
1955 365   254.902   127.629   127.273  1372.195    91.170     8.873   832.747   660.911   156.121    58.930     3.748     0.000
0.000  -0.116435381E-09
1956 366   310.147   281.781    28.367  1903.177    65.457     6.385   828.810   657.786   324.945   -38.969    61.261     8.775
0.000  -0.183099313E-08
1957 365   381.163   363.570    17.592    37.571    15.850     1.742   845.560   671.079   327.175    36.173    16.072     0.000
0.000  -0.249317704E-08
```

1958	365	341.291	301.092	40.198	154.009	29.814	3.133	818.516	649.616	358.730	-51.929	24.106	0.000	0.000
-0.363462775E-08														
1959	365	120.607	110.988	9.619	86.665	14.498	2.372	862.267	684.339	156.361	-30.875	0.000	0.000	0.000
-0.408099911E-08														
1960	366	282.276	271.893	10.384	33.734	8.881	1.503	845.880	671.334	218.626	61.323	0.826	0.000	0.000
-0.741168510E-10														
1962	365	338.712	312.850	25.862	224.619	22.460	3.402	830.644	659.242	267.587	36.307	31.416	0.176	0.000
-0.634508074E-09														
1963	365	233.379	179.558	53.821	1076.487	39.313	9.210	811.546	644.085	260.099	-53.651	12.423	0.000	0.000
-0.143827826E-08														
1964	366	378.187	225.224	152.963	1169.237	47.514	4.296	825.651	655.279	254.243	14.771	3.723	0.000	0.000
-0.201924445E-08														
1966	365	232.387	189.844	42.543	2289.880	128.425	9.780	843.435	669.393	229.022	41.116	48.131	0.000	0.000
-0.403216082E-08														
1967	365	206.302	174.810	31.492	232.016	38.340	3.713	852.245	676.385	279.375	-83.650	17.425	0.000	0.000
-0.648956358E-09														
1968	366	290.905	277.997	12.908	34.296	10.462	2.565	839.402	666.192	247.988	40.471	0.000	0.000	0.000
-0.852496186E-09														
1969	365	333.356	234.815	98.541	1113.232	29.055	4.130	845.652	671.152	264.208	-17.865	17.527	0.000	0.000
-0.138001286E-08														
1970	365	258.175	179.126	79.049	1328.668	124.590	8.793	833.578	661.570	228.047	30.552	45.117	1.920	0.000
-0.350565337E-08														
1971	365	281.384	242.377	39.006	323.340	45.848	4.291	836.134	663.598	291.648	-13.915	10.492	0.000	0.000
-0.431267944E-08														
1974	365	167.323	131.266	36.057	171.037	31.414	4.643	837.322	664.541	187.377	-26.855	2.157	0.000	0.000
-0.259751234E-09														
1975	365	267.498	230.968	36.530	91.871	31.777	4.753	830.917	659.458	262.708	-3.118	3.155	0.000	0.000
-0.708225371E-09														
1976	366	227.329	194.598	32.731	545.162	27.352	5.379	822.342	652.653	241.625	-19.755	0.080	0.000	0.000
-0.103352604E-08														
1977	365	313.618	235.905	77.713	498.200	31.490	3.201	834.295	662.139	230.039	37.357	0.000	0.000	0.000
-0.137197357E-08														
1978	365	257.877	208.165	49.713	721.268	87.363	5.371	849.230	673.992	271.385	-17.794	41.937	0.000	0.000
-0.283669747E-08														
1979	365	313.519	282.240	31.280	67.986	29.239	2.041	843.151	669.167	227.781	69.200	14.497	0.000	0.000
-0.261695151E-09														
1980	366	257.877	237.226	20.651	128.283	18.391	2.150	857.524	680.574	289.409	-69.175	35.383	0.299	0.000
-0.280423179E-08														
1981	365	390.486	276.116	114.370	406.011	72.334	8.066	814.972	646.803	254.882	71.770	21.798	0.000	0.000
-0.366651546E-08														
1982	365	328.694	297.352	31.342	446.031	55.426	4.752	816.770	648.230	346.565	-25.551	31.765	0.000	0.000
-0.594888706E-08														
1983	365	450.889	307.306	143.583	1341.380	70.336	8.617	813.936	645.981	297.566	29.337	50.739	0.464	0.000
-0.843875677E-08														
1987	365	238.933	199.277	39.656	1924.282	94.977	7.509	848.007	673.021	294.840	-52.029	51.443	5.062	0.000
-0.210783399E-08														
1995	365	412.604	368.866	43.738	277.639	46.667	4.115	824.088	654.038	312.516	50.800	52.217	0.237	0.000
-0.439056246E-08														
1996	366	364.301	254.389	109.912	636.803	78.251	7.623	825.734	655.345	312.781	-39.681	59.540	5.863	0.000
-0.763445402E-08														

Global Summary Statistics (mm/year):

Precipitation.....	290.532770
Rain.....	233.901757
Snowfall.....	56.631013
Snow-cover.....	631.284972
Snow-melt.....	50.640619
Sublimation.....	5.261948
Potential Evapotranspiration....	832.586015
Actual Evapotranspiration.....	262.509388
Change in Soil Moisture.....	0.727825
Net Infiltration.....	21.305163
Runoff Generation.....	0.690760
Cumulative Daily Run-on.....	0.094091
Outflow.....	0.000000
Average Mass Balance Error.....	-0.264155E-12
Average Max Daily Error (mm/dy).	-0.386456E-13

Table II-4. Input Control File Used on Platform Pentium Pro PC and Operating System Windows NT 4.0 Using INFIL Version 2.0

```
INFIL2a1.ctl: Solitario East, Tule Lake, se10-2a1-gm1-k2-w20 (11/18/1999)
1   0.00000001                                  IROUT(1 = coupled, 0 = uncoupled, -1 = flow routing off, -2 = infil off), IFRTOL
2   2     1.78                                      ISNOW, ISNWMOD, SNOPAR1
3   0.1   0.3                                   ISUBLIM, SUBPAR1, SUBPAR2
0   1.                                          IPPTTEST (1 for testing), PPTTEST = constant
0   .0                                          IETTEST (1 for testing), ETTEST = constant
30.0                                                CELSIZE (node spacing (meters): using for flow volume calculations)
544721.0  4072203.0                             xcfs,ycfs: coords for discharge cell
1950  1    1996    366   1   5                 yr1 = start year, dnn1 = start day, yr2 = end year, dn2 = end day
1     1    1    1                              multipliers ( pptfact, etfact, imbfact, sksfact)
1.0   0    0.4                                 SDFACT (soil depth multiplier), IVEGC (set to 1 for map data), FVEGC (use if IVEGC =
0)
1.0   0.8   0.8    0.2                          ROOTF1,ROOTF2,ROOTF3,ROOTF4
1.0   1.0   0.5    0.1                          MAXWGT1, MAXWGT2, MAXWGT3, MAXWGT4
0.3   2.0   3.0    2.0                           RDEPTH1,RDEPTH2,RDEPTH3,RDEPTH4,RDEPTHF
0.02  1.0  .25                                  RKPOR, RKMMFACT, FLAREA
1   1    0                                      INFMOD, ETMOD, RUNMOD
 -10.0  1.04                                 BARSOIL1, BARSOIL2:  bare soil et parameters
3   17.3  11.74                                 IAIRTEMP = 1 for new air temp model, ATEMP1 = avg. air temp, ATEMP2 = air temp
seasonal deviation
2                                               HSTEP: time step for PET model (hours)
5  181  3                                       PPTYUC (=5 diminished elev. correlation, =2 for 4JA , =1 for simple elevation
transfer), AAPREPX, IPPTDAT
Tulelake.inp                                    input file name:  daily precip
     0      1    0.5      1     1               dpthflag, irtz, delvwcf, moistcr, fracmod
     0      1.75                                 IVWCFLG,   vwcfact
se10.w20                                         input file name:  map parameters (*.inp)
-1
se10-gm1.2a1                                       dayall: average daily mass balance terms
1    1                                       ndaymap, imap
70   1995
se10-gm1.2a2
se10-gm1.2a3                                       output file name:  daily mass balance
se10-gm1.2a4                                       output file name:  total daily fluxes
1      1                                        dbgflag, dbgflag2
se10-gm1.2a5                                       output file name:  annual mass balance terms
0                                               IDEBUG:  debugging option parameter
se10-gm1.2a6                                       new debug file
se10-gm1                                         map output:  annual totals or mult-year averages
--
--------------- Parameters for dynamic root-zone function -----------------------------------
--     depth  rtza   rtzb   rtzc  rtzd  bsoil    delvwc
--     m
     4
     1    0.5     15      3     3     2     0.30      0.50
     2    1.5     15      4     2     2     0.20      0.50
     3    4.5     10    1.5     1     2     0.10      0.50
     4    6.0     10    1.5     1     2     0.05      0.25
--
```

```
--------------- Soil Properties (Brooks & Corey/van Genuchten combined) ----------------------------------------
--      fdcp   etresidpor    beta  alpah ksat     PE        B     n     vg-alpha sorp     SOILP   potis
--                                       Kg s /m^3J/Kg                  1/(J/Kg) ???              J/Kg
    10
     1  0.242  0.054  0.366  -3.5  1.26  5.6E-04 -1.19E+00  4.72  1.24  5.2E-01    0.390    0.05 -1.0E+02
     2  0.173  0.023  0.315  -3.5  1.26  1.2E-03 -9.41E-01  3.70  1.31  6.2E-01    0.500    0.05 -1.0E+02
     3  0.163  0.017  0.325  -3.5  1.26  1.3E-03 -8.60E-01  3.36  1.36  6.6E-01    0.510    0.05 -1.0E+02
     4  0.073  0.002  0.281  -3.5  1.26  3.8E-03 -6.22E-01  2.18  1.62  8.7E-01    0.700    0.05 -1.0E+02
     5  0.200  0.035  0.330  -3.5  1.26  6.7E-04 -1.07E+00  4.14  1.78  5.6E-01    0.400    0.05 -1.0E+02
     6  0.150  0.011  0.339  -3.5  1.26  2.7E-03 -7.55E-01  3.06  1.40  7.4E-01    0.700    0.05 -1.0E+02
     7  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
     8  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
     9  0.189  0.028  0.322  -3.5  1.26  5.7E-04 -1.08E+00  3.88  1.30  5.5E-01    0.370    0.05 -1.0E+02
    10  0.189  0.028  0.322  -3.5  1.26  5.7E-05 -1.08E+00  3.88  1.00  5.5E-01    0.037    0.05 -1.0E+02
--
---------------Rock1 Properties (Brooks & Corey/van Genuchten combined)----------------------------------------
--      rkcp   etresidpor    beta  alpha ksat     PE        B     n     vg-alpha fracks   imbibe  potir
--                                       Kg s /m^3J/Kg                  1/(J/Kg) Kg s /m^3mm/dy   J/Kg
    129
      1    0.35    0.065    0.366    -1.5    1.26   0.0011    -0.93     3.6    1.25    0.52  0.00056     500    -100
      2   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.41    -100
      3   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
      4   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
      5   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
      6   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.41    -100
      7   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
      8   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
      9   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
     10   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
     11   0.105     0.02     0.11    -1.5    1.26  4.0E-09       -6    3.92    1.47      64  2.4E-07    0.06    -100
     12   0.388    0.039    0.393    -1.5    1.26 0.000053     -242    4.57    1.38    2400 0.000053    2.74    -100
     13    0.43    0.044    0.435    -1.5    1.26   0.0004    -1790    6.81    1.23   18000   0.0004   13.83    -100
     14   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
     15   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
     16   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
     17   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.35    -100
     18   0.248     0.01    0.253    -1.5    1.26  3.8E-06    -82.3    2.71    1.84     830  3.9E-06    3.34    -100
     19   0.159     0.01    0.164    -1.5    1.26  1.2E-06     -140    3.61    1.53    1400  1.3E-06    1.13    -100
     20   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     21   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     22   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     23   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     24   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     25   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     26   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     27   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     28   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     29   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     30   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     31   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     32    0.43    0.044    0.435    -1.5    1.26   0.0004    -1790    6.81    1.23   18000   0.0004   13.83    -100
```

33	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
34	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
35	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
36	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
37	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
38	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
39	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
40	0.494	0.05	0.499	-1.5	1.26	0.000088	-341	4.19	1.43	3400	0.000088	75.62	-100
41	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
42	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
43	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
44	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-26.9	5.6	1.29	270	2.1E-07	0.09	-100
45	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
99	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
46	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
47	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
48	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
49	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
50	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
51	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
52	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
53	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
54	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
55	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
56	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
57	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
58	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
59	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
60	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
61	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
62	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
63	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
64	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
65	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
66	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
67	0.031	0.018	0.036	-1.5	1.26	5.0E-09	-0.6	3.39	1.58	10	2.4E-07	0.17	-100
68	0.168	0.021	0.173	-1.5	1.26	7.3E-08	-125	5.36	1.31	1300	1.3E-07	0.07	-100
69	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.02	-100
70	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
71	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.01	-100
72	0.317	0.023	0.322	-1.5	1.26	3.8E-08	-181	4.01	1.46	1800	9.9E-08	1.65	-100
73	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
74	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
75	0.281	0.051	0.286	-1.5	1.26	1.7E-08	-17.5	4.01	1.45	180	7.8E-08	0.02	-100
76	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
77	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
201	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
202	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.06	-100
203	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
204	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	120	1.4E-07	500	-100
205	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100

206	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
207	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
208	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
209	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
210	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
211	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
212	0	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
213	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
214	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
301	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
302	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
303	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
304	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
305	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
306	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
307	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
308	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
309	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
310	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
311	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
312	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
313	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
314	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
315	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
316	0.159	0.01	0.164	-1.5	1.26	1.2E-06	-140	3.61	1.53	1400	1.3E-06	1.13	-100
317	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
318	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
319	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
320	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
321	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
322	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
323	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
324	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
325	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
326	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
327	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
328	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
329	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
330	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
331	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
332	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
333	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
334	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
335	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
336	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
345	0.097	0.01	0.102	-1.5	1.26	1.3E-07	-154.6	5.02	1.56	1560	2.9E-07	0.24	-100

Table I-5. Summary Output File Produced on Platform Pentium Pro PC PC and Operating System Windows NT 4.0 Using INFIL Version 2.0

```
INFIL2a1.ctl: Solitario East, Tule Lake, se10-2a1-gm1-k2-w20 (11/18/1999)
Daily precipitation input:              Tulelake.inp
Watershed modeling domain parameters:   se10.w20
Average daily mass balance terms:       se10-gm1.2a1
24-hour mass balance map:               se10-gm1.2a2
Annual mass balance map:                se10-gm1.1
Average annual mass balance map:        se10-gm1.2a4
Summary statistics output:              se10-gm1.2a3
Debugging output:                       se10-gm1.2a5


Total number of days read in   =    12053
Total daily precip             =   9660.7
Average annual precip (mm)     =    292.8
Maximum daily precip (mm)      =     51.3


TOTAL NUMBER OF LOCATIONS      =       69
AVERAGE ELEVATION OF SAMPLE    =   1339.3
MAXIMUM ELEVATION OF SAMPLE    =   1377.0
MINIMUM ELEVATION OF SAMPLE    =   1301.0
AVERAGE SOIL DEPTH (M)         =    2.124
AVERAGE SLOPE OF SAMPLE        =     12.8
MAXIMUM SLOPE OF SAMPLE        =     23.0
NUMBER OF ACTIVE LOCATIONS     =       17
```

Yr	Dy	Precip	Rain	Snow	Sn-cover	Snowmelt	Sublim	PET	PETRS	Evapotrs	Del-soil	Net-inf	Runoff	Out-flow	Mass-balance
1950	365	288.823	218.232	70.591	432.356	60.488	5.658	835.723	663.272	242.653	36.067	0.000	0.000	0.000	0.175624822E-09
1951	365	212.451	120.978	91.473	736.625	54.355	7.819	842.857	668.934	194.543	-19.922	0.711	0.000	0.000	0.218118841E-09
1952	366	271.664	232.269	39.395	354.124	64.979	4.581	812.532	644.867	292.275	-8.315	13.289	0.000	0.000	0.192151030E-09
1953	365	367.872	313.935	53.937	178.650	51.584	5.932	799.670	634.659	315.938	37.982	11.599	0.000	0.000	0.263993153E-09
1954	365	212.451	135.956	76.495	495.141	53.006	7.244	813.631	645.739	223.574	-55.092	20.480	0.000	0.000	0.241319896E-09
1955	365	254.902	127.629	127.273	1372.197	91.170	8.873	832.747	660.910	156.121	58.930	3.748	0.000	0.000	-0.112246360E-09
1956	366	310.147	281.781	28.367	1903.180	65.457	6.385	828.810	657.786	324.945	-38.969	61.261	8.775	0.000	-0.156601933E-09
1957	365	381.162	363.570	17.592	37.572	15.850	1.742	845.560	671.079	327.175	36.173	16.072	0.000	0.000	-0.164860299E-09
1958	365	341.291	301.092	40.198	154.009	29.814	3.133	818.516	649.616	358.730	-51.929	24.106	0.000	0.000	0.444154606E-10

1959	365	120.607	110.988	9.619	86.666	14.498	2.372	862.267	684.339	156.361	-30.875	0.000	0.000	0.000
1960	366	282.276	271.893	10.384	33.734	8.881	1.503	845.880	671.334	218.626	61.323	0.826	0.000	0.000
1962	365	338.712	312.850	25.862	224.620	22.460	3.402	830.644	659.241	267.587	36.307	31.416	0.176	0.000
1963	365	233.379	179.558	53.821	1076.488	39.313	9.210	811.546	644.084	260.099	-53.651	12.423	0.000	0.000
1964	366	378.187	225.224	152.963	1169.237	47.514	4.296	825.651	655.279	254.243	14.771	3.723	0.000	0.000
1966	365	232.387	189.844	42.543	2289.883	128.425	9.780	843.435	669.393	229.022	41.116	48.131	0.000	0.000
1967	365	206.302	174.810	31.492	232.017	38.340	3.713	852.245	676.385	279.375	-83.650	17.425	0.000	0.000
1968	366	290.905	277.997	12.908	34.296	10.462	2.565	839.402	666.192	247.988	40.471	0.000	0.000	0.000
1969	365	333.356	234.815	98.541	1113.233	29.055	4.130	845.652	671.152	264.208	-17.865	17.527	0.000	0.000
1970	365	258.175	179.126	79.049	1328.669	124.590	8.793	833.578	661.570	228.047	30.552	45.117	1.920	0.000
1971	365	281.384	242.377	39.006	323.340	45.848	4.291	836.133	663.598	291.648	-13.915	10.492	0.000	0.000
1974	365	167.323	131.266	36.057	171.037	31.414	4.643	837.322	664.541	187.377	-26.855	2.157	0.000	0.000
1975	365	267.498	230.968	36.530	91.871	31.777	4.753	830.917	659.458	262.708	-3.118	3.155	0.000	0.000
1976	366	227.329	194.598	32.731	545.162	27.352	5.379	822.342	652.653	241.625	-19.755	0.080	0.000	0.000
1977	365	313.618	235.905	77.713	498.201	31.490	3.201	834.295	662.139	230.039	37.357	0.000	0.000	0.000
1978	365	257.877	208.165	49.713	721.270	87.363	5.371	849.229	673.992	271.385	-17.794	41.937	0.000	0.000
1979	365	313.519	282.240	31.280	67.986	29.239	2.041	843.151	669.167	227.781	69.200	14.497	0.000	0.000
1980	366	257.877	237.226	20.651	128.283	18.391	2.150	857.524	680.574	289.409	-69.175	35.383	0.299	0.000
1981	365	390.486	276.116	114.370	406.012	72.334	8.066	814.972	646.803	254.882	71.770	21.798	0.000	0.000
1982	365	328.694	297.352	31.342	446.031	55.426	4.752	816.770	648.230	346.565	-25.551	31.765	0.000	0.000
1983	365	450.889	307.306	143.583	1341.381	70.336	8.617	813.935	645.981	297.566	29.337	50.739	0.464	0.000
1987	365	238.933	199.277	39.656	1924.285	94.977	7.509	848.007	673.021	294.840	-52.029	51.443	5.062	0.000
1995	365	412.604	368.866	43.738	277.639	46.667	4.115	824.088	654.038	312.516	50.800	52.217	0.237	0.000
1996	366	364.301	254.389	109.912	636.804	78.251	7.623	825.734	655.345	312.781	-39.681	59.540	5.863	0.000

0.187447951E-09	0.364036219E-10	0.156433114E-09	0.137249349E-09
0.531510373E-10	0.619987124E-10	0.923037274E-10	0.614522354E-10
0.656519736E-10	0.883941765E-10	0.168360502E-09	0.687992874E-10
0.182410852E-09	0.218551604E-09	0.180917354E-09	0.114093109E-09
0.100798462E-09	0.193833364E-09	0.557352754E-10	0.160379148E-09
0.492668107E-09	0.710995995E-10	0.251104930E-09	0.259870723E-09

Global Summary Statistics (mm/year):

```
-----------------------------------------------

Precipitation..................        290.532762
Rain...........................        233.901750
Snowfall.......................         56.631012
Snow-cover.....................        631.285751
Snow-melt......................         50.640612
Sublimation....................          5.261954
Potential Evapotranspiration....       832.585923
Actual Evapotranspiration.......       262.509371
Change in Soil Moisture........          0.727825
Net Infiltration...............         21.305166
Runoff Generation..............          0.690760
Cumulative Daily Run-on........          0.094091
Outflow........................          0.000000
Average Mass Balance Error......    -0.998939E-13
Average Max Daily Error (mm/dy).    -0.109569E-13
```

ATTACHMENT II

LISTING OF SOURCE CODE

FOR

INFIL V A_2a.1

This attachment contains a listing of INFIL (INFIL V A_2a.1) that was used on the DEC ALPHA platform at Sandia National Laboratories in this analysis. The use of this code for this analysis is as unqualified. A Software Activity Plan, Software Users Request, and a copy of the executable has been submitted to the Software Secretariat.

```
1          program infil2a1
2    c     compiled 7/26/99, p944dcaser, WinNT
3    c
4    c     program INFIL, version 2.a1
5    c     developers Joe Hevesi, Alan Flint, U.S Geological Survey
6    c     final version of test program infils5o
7    c*********************************************************************
8    **
9    c     modified from program INFIL, version A_2.a1
10   c       modified for DEC ALPHA by Ron McCurley beginning July 27, 1999
11   c         - added "status=unknown" to open statements
12   c         - modified OPEN statements with max record length
13   modifications
14   c           (i.e. 'recl=256' or 'recl=512')
15   c       commented out writes to unnecesary auxillary output files to
16   c       facilitate quicker run times (units 13, 14, *)
17   c*********************************************************************
18   **
19   c This is a primary tree starting at the program 'INFIL2A1'
20   c
21   c INFIL2A1-+-(ADJUSTR)
22   c          |
23   c          +-(ADJUSTL)
24   c          |
25   c          +-(ADJUSTL)
26   c          |
27   c          +-ATMOS
28   c          |
29   c          +-POTEVAP
30   c          |
31   c          +-PPTDIST
32   c          |
33   c          +-SNOW
34   c          |
35   c          +-V4CBUCK
36   c          |
37   c          +-IFR
38   c
39   c
40   c     max dimension = 100,000 grid cells
41   c     max simulation time = 400,000 days (1,095.14 years)
42   c
43   c
44         integer itime,ipptdat,ia,ndays2,ncell
45         integer rown,coln,edge,n2,imap
46         integer irday(10),nyrout(10)
47   c
48         integer nrockid,nsoilid,rockid,soilid
49         integer maxiter,maxit,minit,iter
50         integer i,dbgflag,ilocout,ndays,ndayout,depthflg,irtz,moistcr
51         integer yrbeg,yrend,dnbeg,dnend,nall,tyear1,tyear2
52         integer pptyuc,dbgflag2,locstart,lnumber,inumber
53   c
54   c---- temporal input file parameters
55   c     ppt = daily precip (mm)
56   c     maxairt = max daily air temp (deg C)
57   c     minairt = min daily air temp (deg C)
```

```
58   c      avgairt = avg daily air temp (deg C)
59   c
60          integer yr(400000),dn(400000),month(400000),day(400000)
61          double precision ppt(400000),maxairt(400000),
62         1            minairt(400000),avgairt(400000)
63          real snowdat
64   c
65   c---- variables for reading in quality flags
66   c      and station ID info included with climate input
67          character*2 dayflg
68          character*6 stcode
69          character*8 stname
70   c
71   c---- overland flow routing parameters
72          integer row(100000),col(100000),hole(100000),iwat(100000)
73          real elev(0:800,0:800)
74          double precision flowout(0:800,0:800)
75          double precision totout(0:800,0:800),totin(0:800,0:800)
76          double precision flowin(300000)
77   C
78   C---- map parameter input file variables
79          integer locid(100000),topoid(100000),
80         1        soiltype(100000),rocktype(100000)
81   c
82          double precision easting(100000),northing(100000),
83         1                 sl(100000),asp(100000)
84          double precision ridge(100000,0:36)
85          double precision lat(100000),lon(100000)
86          double precision viewfactor
87          double precision depth(100000)
88   C
89          real years
90          double precision tpet,petrs
91          double precision ssw2,avgsw2
92   c
93          double precision sbalance
94          double precision SSPPT,AVGPPT2,SSET,AVGET,SSPOND,AVGPOND
95   C
96          double precision PONDFACT,FR2FACT
97          double precision MAXPPT,SPPT,AVGELEV,MAXELEV,MINELEV,SELEV
98          double precision AVGSL,SSL,MAXSL
99          double precision pptfact2,pptfact,etfact,avgppt
100  C
101         DOUBLE PRECISION BETA(366),PG(366),WP(366),CSR(366),OZONE(366)
102         double precision TR,RD,DR,PI
103  C
104  c----- Richards eq. parameters for rock and soil types ---------
105  c       max of 300 different rock types, 20 different soil types
106         double precision ROCKCAP(0:500),ROCKRESID(0:500),ROCKPORO(0:500),
107        1     ROCKALBETA(0:500),ROCKTALPHA(0:500),ROCKKS(0:500),
108        2     ROCKPE(0:500),ROCKBVAL(0:500),ralpha(0:500),
109        3     ROCKFRACKS(0:500),POTIR(0:500),ROCKVGN(0:500),imbibe(0:500)
110
111         double precision
112  FIELDCAP(20),SOILRESID(20),SOILPORO(20),sorp(20),
113        1     SOILALBETA(20),SOILTALPHA(20),POTIS(20),salpha(20),
114        2     SOILKS(20),SOILPE(20),SOILBVAL(20),SOILPOND(20),SOILVGN(20)
```

```
115   c
116   c
117   c---- Dynamic root zone parameters ----------------------------
118   c
119         integer IDPTH
120         double precision RTZA(10),RTZB(10),RTZC(10),
121        1                  RTZD(10),BSOIL(10),VWCF2(10)
122         double precision IDPTH2
123         double precision SRTZFN,BSOIL2,DELVWCF
124   c
125   c---- Thomas Algorithm parameters (after G. Camplbell) -------------
126         double precision wd,gr,im2,im3
127   c
128   c---- following parameters are specific to the BUCKET model
129         double precision soilvwc(100000),runoffmm(100000),
130        1                 infilmm(100000)
131         double precision massbal,finalvwc
132         double precision imb,tet,dsoilmm
133         double precision totppt,tottet,totdsw,totnet,totinf,totoff,
134        1           totpet,totbal
135         double precision zdum(10)
136         double precision z7,vwcfact
137         integer locid2,year2
138         double precision pptday,tetday,infday,runday,
139        1                  petday,petrsday,atday,soilday
140   c
141   c
142   c---- new parameters
143   c
144   c     vegt = vegetation type
145   c     vegc = vegetation cover
146   c     isoild = variable soil depth option
147   c
148         integer vegt,isoild,noutyear,stepyear
149         double precision totoutc
150         double precision vegc(100000)
151         double precision imbfact
152   c
153   c     terms for annual summary statistics
154         double precision pptyr,tetyr,dswyr,infyr,runyr,
155        1                  outyr,petyr,petrsyr,atyr
156         double precision airtemp
157   c
158   c---- mapping terms
159   c     (year = annual, aa = average annual, gaa = global)
160         double precision yearppt(100000),yeartet(100000),
161        1      yeardsw(100000),yearnet(100000),yearinf(100000),
162        2      yearoff(100000),yearpet(100000),yearbal(100000),
163        3      yearrun(100000)
164   c
165         double precision aappt(100000),aapet(100000),aatet(100000),
166        1      aadsw(100000),aainf(100000),aaoff(100000),aabal(100000),
167        2      aarun(100000),aanet(100000),dayvwc(100000)
168   c
169         double precision gaappt,gaapet,gaatet,gaadsw,gaainf,
170        1            gaaoff,gaarun,gaanet,gaamb1,gaamb2
171   c
```

```
172          double precision sdfact,avgdepth,atemp1,atemp2,ta
173    c
174    c---- new 7/12/98 (infilsda)
175          integer dclass,iairtemp,ibuck2
176          double precision fdepth
177          double precision rockpor(100000),rockmm(100000)
178          double precision cdepth(4,100000)
179          double precision soilmm(3,100000)
180          double precision pptloc(100000)
181    c
182    c---- added 3/29/99 (infils5d): snow-cover module terms
183          double precision snowmm(100000),snowfall(100000),
184         1                 rain(100000),sublim(100000)
185          double precision yearrain(100000),yearsnow(100000),
186         1                 yearsncv(100000),yearmelt(100000),
187         +                 yearsubl(100000)
188          double precision aarain(100000),aasnow(100000),aasncv(100000),
189         1                      aamelt(100000),aasubl(100000)
190          double precision rainyr,snowyr,sncvyr,meltyr,sublyr
191          double precision melt,meltday,snowday,rainday,sncvday,sublday
192          double precision gaarain,gaasnow,gaasncv,gaamelt,gaasubl
193          double precision massb2(100000)
194          double precision dsnowcv
195          double precision trn,petrs2
196          integer isnow,isnwmod,isublim
197          real snopar1,subpar1,subpar2
198    c
199          double precision xcfs,ycfs,cfsday,totout3,totout2,totflow
200          integer idebug,dayn,infmod,etmod,runmod
201          double precision rootf1,rootf2,rootf3,rootf4
202          double precision ifrtol,rkpor,flarea
203          double precision barsoil1,barsoil2,rdepthf
204          double precision rdepth1,rdepth2,rdepth3,rdepth4,rkmmfact
205          real hstep
206          double precision maxbal(100000),massb(100000)
207          double precision rinfmm1(100000),rinfmm2(100000)
208          double precision yearinfr(100000),aainfr(100000)
209          double precision ppttest,ettest,tsoilmm
210          double precision rinfday,rinfday2,infrday
211          double precision sksfact,imb2,skmm,fnd,fvegc
212          double precision maxwgt1,maxwgt2,maxwgt3,maxwgt4,massbal2
213
214          integer ippttest,iettest,ivegc
215
216          real aaprepx
217          double precision celsize
218          double precision daybal,locbal,ndays5
219
220          integer ndays4
221          integer icalday
222          integer optmassb
223    c
224    c---- file names
225          character*20 infile,outfile,pptfile,dbgfile,flxfile
226          character*20 dayall,vwcfile,outyear3,dbugout
227          character*20 daymap(10)
228          character*8  outyear1
```

```
229         character*3  outyear2
230         character*20 outyear(2000)
231         character*256 header,header0,header1
232   C
233   c
234   c==================================================================
235   c=============== begin main program ==============================
236   c
237   c     open control file and all I/O files
238   c
239    5    format(A)
240    6    format(1x,a100)
241   c
242
243   open(unit=7,file='infil2a1.ctl',form='formatted',status='unknown')
244         read(7,5) header
245   c
246   c     hard-wired optmassb (infil2a1, 7/23/99)
247   c     read(7,*) optmassb
248   c     optmassb = 1
249         read(7,*) irout, ifrtol
250   c
251   c     optmassb = 1 to stop runoff routing on last day of simulation
252   c     irout = 0 for de-coupled flow routing, = 1 for coupled routing
253   c     irout = -1 for no flow routing, -2 for no infiltration
254   c     ifrtol = flow routing tolerance term
255   c
256         read(7,*) isnow,isnwmod,snopar1
257         read(7,*) isublim,subpar1,subpar2
258         read(7,*) ippttest,ppttest
259         read(7,*) iettest,ettest
260         read(7,*) celsize
261   c
262   c     ippttest = 1 sets precip to constant = ppttest
263   c     iettest = 1 sets et to constant = ettest
264   c
265   c     isnow = 2 : snow module enabled
266   c     isnow = 0 : snow module dissabled
267   c     isnwmod = snow-cover model type
268   c                       >= 1 for setting et = 0 for days at freezing or
269   below
270   c                       = 2 for snow-cover mod enabled
271   c     snopar1 = model parameter
272   c
273   c     isublim = sublimation model option (0 = off, 1 = on)
274   c     subpar1 = sublimation factor
275   c
276   c     added input for generating discharge output
277   c     at user specified location 7/21/98
278         read(7,*) xcfs,ycfs
279   c
280   c     tyear option added 5/2/97 for limiting output
281   c     tyear1: controls output to screen
282   c     tyear2: controls output to file
283   c
284         read(7,*) yrbeg,dnbeg,yrend,dnend,tyear1,tyear2
285   c     read(7,*) maxiter,maxdt,mindt,maxit,minit,hstep
```

```
286   c      read(7,*) ilocout,ndayout
287   c
288   c      pptfact = daily precip scaling factor
289   c      etfact  = daily potential et scaling factor
290   c      imbfact = bedrock permeability (mm/day) scaling factor
291   c      sksfact = soil permeability scaling factor
292          read(7,*) pptfact,etfact,imbfact,sksfact
293   c
294   c      sdfact = soil depth scaling factor
295   c      rootf1 = upper cascading bucket et weighting function limit
296   c      rootf2 = 2nd cascading bucket " " " limit
297   c      rootf3 = 3rd cascading bucket " " " limit
298   c      rootf4 = lower cascading bucket " " " limit
299          read(7,*) sdfact,ivegc,fvegc
300          read(7,*) rootf1,rootf2,rootf3,rootf4
301          read(7,*) maxwgt1,maxwgt2,maxwgt3,maxwgt4
302   c
303          read(7,*) rdepth1,rdepth2,rdepth4,rdepthf
304          read(7,*) rkpor,rkmmfact,flarea
305          read(7,*) infmod,etmod,runmod
306   c
307   c      bares1 = barsoil1 soil et parameter 1
308   c      bares2 = barsoil2 soil et parameter 2
309   c      iairtemp = airtemp model type (0 = original, 1 = new)
310   c      atemp1   = average annual air temp (deg. C)
311   c      atemp2   = half amplitude of air temp deviation (deg. C)
312   c      hstep  = time step for modeling PET (hours)
313          read(7,*) barsoil1,barsoil2
314          read(7,*) iairtemp,atemp1,atemp2
315          read(7,*) hstep
316   c
317   c      pptyuc = precip spatial distribution model
318   c      aaprepx = average annual precip for site (use when pptyuc = 5)
319   c      ipptdat = input data type: 1 = stochastic simulation, 0 = data
320   c      pptfile (unit 11) = daily precip input file
321          read(7,*) pptyuc,aaprepx,ipptdat
322          read(7,5) pptfile
323   c
324          read(7,*) depthflg,irtz,delvwcf,moistcr,fracmod
325          read(7,*) ivwcflg,vwcfact
326          if(ivwcflg.eq.1) then
327             read(7,5) vwcfile
328             open(unit=16,file=vwcfile,status='unknown')
329             do II = 1,12
330                read(16,5) header0
331             enddo
332          endif
333   c
334   c      infile (unit 8) = main input file: geospatial input parameters
335          read(7,5) infile
336   c
337   c      option to re-start after crash (4/30/97)
338          read(7,*) locstart
339   c
340   c      dayall (unit 14) = daily mass balance terms averaged for all
341   points
342   c      same as screen output
```

```
343          read(7,5) Dayall
344    c
345    c     daymap(i) (unit 10) = 24-hour mass balance results for mapping
346    c     maximum 10 output files
347          read(7,*) ndaymap, imap
348          do i = 1,ndaymap
349                read(7,*) irday(i),nyrout(i)
350                read(7,5) daymap(i)
351          enddo
352    c
353    c     outfile (unit 9) = summary info and annual summary statistics
354          read(7,5) outfile
355    c
356    c     flxfile (unit 12) = average annual mass balance terms
357          read(7,5) flxfile
358    c
359    c     dbgfile (unit 13) = 2nd daily mass balance output file
360          read(7,*) dbgflag,dbgflag2
361          read(7,5) dbgfile
362    c
363    c---- added idebug option 8/2/98 to check mass balance
364    c     if idebuh = 0 then echo input parameters
365          read(7,*) idebug
366          read(7,5) dbugout
367          open(unit=23,file=dbugout,status='unknown',recl=256)
368    c
369    c---- noutyear = number of annual map files
370    c     or multi-year average map files (outyear(i) = unit 15)
371    c     mod 3/19/1999 to handle max of 1000 annual maps (INFILS5B)
372    c     ------------------------------------------------------------
373          read(7,*) outyear1
374          outyear1 = adjustr(outyear1)
375
376          if(tyear1.eq.1) then
377                stepyear = tyear2
378          else
379                stepyear = 1
380          endif
381
382          open(unit=17,file='yearfile.txt',status='unknown')
383          noutyear = 0
384
385          do i = yrbeg,yrend,stepyear
386                noutyear = noutyear + 1
387                noutyrf = noutyear
388                if(noutyear.gt.999) noutyrf = 0
389                write(17,10) noutyrf
390    10          format(i3)
391          enddo
392    c
393          rewind(17)
394
395          do i = 1,noutyear
396                read(17,5) outyear2
397                outyear2 = adjustl(outyear2)
398                outyear(i) = (outyear1//'.'//outyear2)
399    c
```

```
400          enddo
401
402          rewind(17)
403
404          do i = 1,noutyear
405                write(17,5) outyear(i)
406          enddo
407
408          close(17)
409    c
410    C
411    C---- read in parameters for dynamic root-zone function -------
412          do i = 1,4
413             read(7,5) header0
414          enddo
415
416          read(7,*) NUMDEPTH
417
418          do I = 1,NUMDEPTH
419             READ(7,*) IDPTH,IDPTH2,RTZA(IDPTH),RTZB(IDPTH),
420         1             RTZC(IDPTH),RTZD(IDPTH),BSOIL(IDPTH),
421         2             VWCF2(IDPTH)
422          enddo
423    C
424    C---- read in soil properties data -----------
425          do i = 1,4
426             READ(7,5) header1
427          enddo
428
429          READ(7,*) NSOILID
430
431          DO I = 1,NSOILID
432             READ(7,*) SOILID,FIELDCAP(SOILID),SOILRESID(SOILID),
433         1             SOILPORO(SOILID),SOILALBETA(SOILID),
434         2             SOILTALPHA(SOILID),SOILKS(SOILID),SOILPE(SOILID),
435         3             SOILBVAL(SOILID),SALPHA(soilid),SOILVGN(SOILID),
436         4             SORP(SOILID),soilpond(soilid),POTIS(SOILID)
437          ENDDO
438    C
439    C---- read in rock properties data -----------
440
441          do i = 1,4
442             READ(7,5) header1
443          enddo
444
445          READ(7,*) NROCKID
446
447          DO I = 1,NROCKID
448             READ(7,*) ROCKID,ROCKCAP(ROCKID),ROCKRESID(ROCKID),
449         1                             ROCKPORO(ROCKID),ROCKALBETA(ROCKID),
450         2             ROCKTALPHA(ROCKID),ROCKKS(ROCKID),ROCKPE(ROCKID),
451         3             ROCKBVAL(ROCKID),RALPHA(rockid),ROCKVGN(ROCKID),
452         4             ROCKFRACKS(ROCKID),IMBIBE(ROCKID),POTIR(ROCKID)
453          ENDDO
454    C
455    c
456    C---- finished reading control file, write info to summary file
```

```
457   c     ------------------------------------------------------------
458         open(unit=9,file=outfile,status='unknown',recl=256)
459         write(9,6) header
460         write(*,6) header
461   c
462         write(9,15) pptfile
463         write(*,15) pptfile
464   15    format(1x, 'Daily precipitation input:              ',a20)
465   c
466         write(9,16) infile
467         write(*,16) infile
468   16    format(1x, 'Watershed modeling domain parameters:  ',a20)
469   c
470         write(9,17) dayall
471         write(*,17) dayall
472   17    format(1x, 'Average daily mass balance terms:      ',a20)
473   c
474         do i = 1,ndaymap
475               write(9,18) daymap(i)
476               write(*,18) daymap(i)
477   18    format(1x, '24-hour mass balance map:              ',a20)
478         enddo
479   c
480         outyear3 = adjustl(outyear(1))
481         write(9,19) outyear3
482         write(*,19) outyear3
483   19    format(1x, 'Annual mass balance map:               ',a20)
484   c
485         write(9,20) flxfile
486         write(*,20) flxfile
487   20    format(1x, 'Average annual mass balance map:       ',a20)
488   c
489         write(9,21) outfile
490         write(*,21) outfile
491   21    format(1x, 'Summary statistics output:             ',a20)
492   c
493         write(9,22) dbgfile
494         write(*,22) dbgfile
495   22    format(1x, 'Debugging output:                      ',a20)
496   c
497   c     dbgfile = debugging output
498         open(unit=13,file=dbgfile,status='unknown')
499   c
500   C
501   c---- read precipitation file
502   c     ND = total number of days for simulation
503   c     IPPTDAT = 1 for simulated precip input
504   c     IPPTDAT = 2 for precip input as data
505   c     PPTFILE = daily precipitation input file
506   c     mod 4/15/99 (infils5i - infils5j)
507   c     mod 3/18/99 (infils4i - infils5b)
508   c
509         open(unit=11,file=pptfile,status='unknown')
510         if(ipptdat.eq.3) then
511               do i = 1,33
512                     read(11,5) header0
513               enddo
```

```
514          endif
515          nd = 0
516          sppt = 0.
517          maxppt = 0.
518   80        nd = nd +1
519   c
520   c            ipptdat = 3 for reading output generated by DAILY09
521                if(ipptdat.eq.3) then
522                  read(11,8011,end=90) icalday,yr(nd),month(nd),day(nd),
523        1               dn(nd),ppt(nd),dayflg,maxairt(nd),
524        2               dayflg,minairt(nd),dayflg,avgairt(nd),
525        3               dayflg,snowdat,dayflg
526   c
527   8011           format(i8,i5,i3,i3,i5,6(f7.1,a3))
528   c
529   c              ipptdat = 2 for new daily climate output from DAYINP10
530                else if(ipptdat.eq.2) then
531                      read(11,8021,end=90) stname,stcode,yr(nd),month(nd),
532        1               day(nd),dn(nd),ppt(nd),dayflg,maxairt(nd),
533        2               dayflg,minairt(nd),dayflg,avgairt(nd),
534        3               dayflg
535   c
536   8021           format(1x,a8,1x,a6,1x,i4,1x,i2,1x,i2,1x,i3,1x,f5.1,1x,
537        1                     a2,1x,f5.1,1x,a2,
538        2                   1x,f5.1,1x,a2,1x,f5.1,1x,a2)
539   c
540   c              ipptdat = 1 for output from PPTSIM
541                else if(ipptdat.eq.1) then
542                      read(11,*,end=90) month(nd),day(nd),yr(nd),dn(nd),
543        +                                ppt(nd)
544   c
545   c            ipptdat = 0 for reading mod3-ppt.dat
546                    else
547                      read(11,*,end=90) yr(nd),dn(nd),ppt(nd)
548                  yr(nd) = 1900 + yr(nd)
549                  month(nd) = -9
550                  day(nd) = -9
551                  endif
552   C
553                if((yr(nd).ge.yrend).and.(dn(nd).ge.dnend)) then
554                   nd = nd + 1
555                   goto 90
556                endif
557              if(yr(nd).lt.yrbeg) then
558                   nd = nd - 1
559                   goto 80
560                endif
561              if((yr(nd).eq.yrbeg).and.(dn(nd).lt.dnbeg)) then
562                   nd = nd - 1
563                   goto 80
564              endif
565              sppt = sppt + ppt(nd)
566             if(maxppt.lt.ppt(nd)) maxppt = ppt(nd)
567   c
568             goto 80
569   c
570    90    nd = nd - 1
```

```
c
      years = float(nd)/365.25
      avgppt = sppt/years
      write(*,95) nd,sppt,avgppt,maxppt
      write(9,95) nd,sppt,avgppt,maxppt
c
 95   format(//1X,'Total number of days read in   = ',I8,
     2        /1X,'Total daily precip             = ',F8.1,
     3        /1X,'Average annual precip (mm)     = ',F8.1,
     4        /1X,'Maximum daily precip (mm)      = ',F8.1/)
C
c
      nloc    = 1
      ssl     = 0.
      selev   = 0.
      maxsl   = -99999.
      maxelev = -99999.
      minelev =  99999.
      illopx = 0
      lnumber = 0
      inumber = 0
      maxrow  = 0
      maxcol  = 0
c
c
C---- INFILE  = site specific parameter input file
c     read in INFILE for parameters needed to define material
c     properties for Richards Eq and Et function, for
c     net radiation calculations, and for the BUCKET model
c     (either as neutron borehole data or for generating maps)
c     added depthclass parameter (dclass) as dummy input  7/12/98
c
      ncell = 0
      n2    = 0
      elold = 999999
      n = 1
      avgdepth = 0.
c
      open(unit=8,file=infile,status='unknown')
100   read(8,*,end=130) locid(n),easting(n),northing(n),lat(n),lon(n),
      1
      row(n),col(n),iwat(n),totoutc,elev(row(n),col(n)),
     3         sl(n),asp(n),soiltype(n),
     4         dclass,depth(n),rocktype(n),topoid(n),vegt,vegc(n),
     5         (ridge(n,j), j=1,36)
c
            if(iwat(n).ne.-3) ncell = ncell + 1
            n2 = n2+1
            if(elev(row(n),col(n)).gt.elold) then
               write(*,106) n2,elev(row(n),col(n)),elold
106            format(1x,'error in sequence: n2 = ',i8,'elev = ',
     1               f8.2,'previous = ',f8.2)
C WARNING!!!
               stop
            endif
            elold = elev(row(n),col(n))
            hole(n) = 0
```

```
628   c
629                 if(row(n).gt.maxrow) maxrow = row(n)
630                 if(col(n).gt.maxcol) maxcol = col(n)
631   c
632   c             catch error in input sequence (not used anymore)
633
634                 if(locid(n2).lt.locstart) goto 100
635   C
636   c             modify input parameters as needed
637                 vegc(n)  = vegc(n)/100.
638                 depth(n) = depth(n)*sdfact
639                 if(lon(n).lt.0) lon(n) = -lon(n)
640   c
641   c             calculate statistics for watershed domain
642                 if(iwat(n).ne.-3) then
643                       avgdepth = avgdepth + depth(n)
644                       ssl   = ssl + sl(n)
645                       selev = selev + elev(row(n),col(n))
646                       if (maxsl.le.sl(n)) maxsl = sl(n)
647                       if (maxelev.le.elev(row(n),col(n)))
648         1             maxelev = elev(row(n),col(n))
649                       if (minelev.ge.elev(row(n),col(n)))
650         1             minelev = elev(row(n),col(n))
651                 endif
652   c
653   c
654   c-------      set up cascading bucket layer depths
655   c             cdepth(4,n) = thickness of bedrock storage term
656   c             mod 08/28/98 INFILS4D
657   c             cdepth(1,n) = thickness of surface layer for bare-soil E
658   c             cdepth(2,n) = thickness of primary root zone
659   c             cdepth(3,n) = thickness of deep root-zone
660   c             cdepth(4,n) = thickness of bedrock root-zone
661   c
662                 if(rdepthf.ne.0.)
663         1             cdepth(4,n) = rdepth4-(depth(n)/rdepthf)
664                 if(cdepth(4,n).lt.0.) cdepth(4,n) = 0.
665   c
666                 if(depth(n).le.rdepth1) then
667                       cdepth(1,n) = depth(n)
668                       cdepth(2,n) = 0.
669                       cdepth(3,n) = 0.
670   c
671                 else if (depth(n).gt.rdepth1.and.depth(n).le.rdepth2) then
672                       cdepth(1,n) = rdepth1
673                       cdepth(2,n) = depth(n)-rdepth1
674                       cdepth(3,n) = 0.
675   c
676                 else if (depth(n).gt.rdepth2) then
677                       cdepth(1,n) = rdepth1
678                       cdepth(2,n) = rdepth2-rdepth1
679                       cdepth(3,n) = depth(n)-rdepth2
680   c
681                 endif
682   c
683                 n = n + 1
684   c
```

```
685          goto 100
686    c
687    c
688    130   n = n - 1
689          rown = maxrow
690          coln = maxcol
691          nloc    = n
692          avgsl   = ssl/ncell
693          avgelev = selev/ncell
694          avgdepth = avgdepth/ncell
695    C
696          write(*,135) NLOC,AVGELEV,MAXELEV,MINELEV,avgdepth,
697          1                      AVGSL,MAXSL,NCELL
698          write(9,135) NLOC,AVGELEV,MAXELEV,MINELEV,avgdepth,
699          1                      AVGSL,MAXSL,NCELL
700     135  format(/1X,'TOTAL NUMBER OF LOCATIONS    = ',I8,
701         1       /1X,'AVERAGE ELEVATION OF SAMPLE  = ',F8.1,
702         2       /1X,'MAXIMUM ELEVATION OF SAMPLE  = ',F8.1,
703         3       /1X,'MINIMUM ELEVATION OF SAMPLE  = ',F8.1,
704          4       /1x,'AVERAGE SOIL DEPTH (M)       = ',f8.3,
705         5       /1X,'AVERAGE SLOPE OF SAMPLE      = ',F8.1,
706         6       /1X,'MAXIMUM SLOPE OF SAMPLE      = ',F8.1,
707         7       /1X,'NUMBER OF ACTIVE LOCATIONS   = ',I8,//)
708    c
709    c     pause
710    c
711    c     write header line for annual summary output
712          write(9,141)
713    141   format(3x,'Yr',2x,'Dy',4x,'Precip',6x,'Rain',6x,'Snow',
714         1             2x,'Sn-cover',2x,'Snowmelt',4x,'Sublim',7x,'PET',
715         2             5x,'PETRS',2x,'Evapotrs',2x,'Del-soil',
716         3             3x,'Net-inf',4x,'Runoff',2x,'Out-flow',
717         4             4x,'Mass-balance')
718    c
719    C
720    C------ Added VWCFILE for inputting finalvwc from
721    c       previous simulation as initial conditions for
722    c       next simulation
723    c       Format for VWCFILE (*.dx1) is:
724    c          zz(1) = ppt
725    c          zz(2) =
726    C
727    c**** modified following on 4/30/97 to allow re-start option
728          if(ivwcflg.eq.1) then
729             read(16,*) locid2,year2,(zdum(jj),jj=1,6),z7
730              if(locid2.ne.locid(nloc)) then
731                   write(*,137) locid2,locid(nloc)
732     137          format(/1x,'WARNING!!! locid2= ',i8,2x,'locid= ',i8/)
733              endif
734          endif
735    C
736    c     -------------------------------
737    c==== Finished reading all input files ============================
738    c     -------------------------------
739    c
740    c---- initialize constants
741    c
```

```
742   C     DR  = degrees to radians conversion
743   c     RD  = radians to degrees conversion
744   c     GR  = gravitational accelleration (m/s)
745   c     PI  = pie
746   c     WD  = water mass to volume conversion term (water density factor)
747   c
748         dr    = 0.0174533
749         rd    = 57.29579
750         pi    = 3.141592654
751         gr    = 9.8
752         wd    = 1000.
753   c
754   c
755   c---- initialize terms
756   c
757         tpet         = 0.
758         tet          = 0.
759         dsoilmm      = 0.
760         massbal      = 0.
761         totppt = 0.
762         tottet = 0.
763         totdsw = 0.
764         totnet = 0.
765         totinf = 0.
766         totoff = 0.
767         totpet = 0.
768         totbal = 0.
769   c
770         sbalance = 0.
771         noutyr  = 0
772   c
773         pptyr  = 0.
774         rainyr = 0.
775         snowyr = 0.
776         sncvyr = 0.
777         meltyr = 0.
778         sublyr = 0.
779         atyr  = 0.
780         petyr  = 0.
781         petrsyr = 0.
782         tetyr  = 0.
783         dswyr  = 0.
784         infyr  = 0.
785         rinfyr = 0.
786         runyr  = 0.
787         outyr  = 0.
788   c
789         pptday  = 0.
790         rainday = 0.
791         snowday = 0.
792         sncvday = 0.
793         meltday = 0.
794         sublday = 0.
795         atday   = 0.
796         tetday  = 0.
797         petday  = 0.
798         petrsday = 0.
```

```
799          soilday  = 0.
800          infday   = 0.
801          runday   = 0.
802          rinfday  = 0.
803          rinfday2 = 0.
804          runinf   = 0.
805          infrday  = 0.
806
807    c
808    c     -------------------------------------------------------
809    C==== Loop through all locations for initializing parameters
810    ===========
811    C     -------------------------------------------------------
812    C
813          do 150 ia = 1,nloc
814    C
815    c           initialize water balance terms
816                pptloc(ia)    = 0.
817                infilmm(ia)   = 0.
818                runoffmm(ia)  = 0.
819                rockmm(ia)    = 0.
820                snowmm(ia)    = 0.
821                rain(ia)      = 0.
822                snowfall(ia)  = 0.
823                sublim(ia)    = 0.
824                maxbal(ia)    = 0.
825    c
826    c           initial vwc determined by residual VWC and vwcfact
827                soilvwc(ia)   = soilresid(soiltype(ia))*vwcfact
828                if((ivwcflg.eq.1).and.(locid(ia).eq.locid2))
829         1       soilvwc(ia) = z7
830    c
831    c           set initial conditions for 3 soil layers
832    c           in cascading bucket (v4cbuck)
833                do i = 1,3
834                      soilmm(i,ia) = soilvwc(ia)*cdepth(i,ia)*1000.
835                enddo
836
837    c           generate soil conductivities for mapping
838    c           (conversion from kg sec / m^3 to mm/day
839                skmm = soilks(soiltype(ia))*86400*9.8
840                imb2 = imbibe(rocktype(ia))*365.25
841    c
842    c           calculate storage capacities for output
843                scapmm1 = (soilporo(soiltype(ia))
844         1            -soilresid(soiltype(ia)))*cdepth(1,ia)*1000.
845                scapmm2 = (soilporo(soiltype(ia))
846         1            -soilresid(soiltype(ia)))*cdepth(2,ia)*1000.
847                scapmm3 = (soilporo(soiltype(ia))
848         1            -soilresid(soiltype(ia)))*cdepth(3,ia)*1000.
849                rkmm = rkpor*cdepth(4,ia)*1000.
850                scapmmt = scapmm1 + scapmm2 + scapmm3 + rkmm
851    c
852    c
853    c           generate map file to check watershed parameters
854    c           SCAPMMT = total water storage capacity
855                if(idebug.eq.0) then
```

```
856                 if(iwat(ia).ne.-3) then
857                   write(23,107) easting(ia),northing(ia),ia,
858       1                 row(ia),col(ia),elev(row(ia),col(ia)),sl(ia),
859       2
860  depth(ia),cdepth(1,ia),cdepth(2,ia),cdepth(3,ia),
861       3                 cdepth(4,ia),soiltype(ia),skmm,
862       +               soilresid(soiltype(ia)),
863       4                 fieldcap(soiltype(ia)),soilporo(soiltype(ia)),
864       5                 scapmm1,scapmm2,scapmm3,rkmm,scapmmt,
865       6                 rocktype(ia),imbibe(rocktype(ia)),imb2
866  c
867  107            format(1x,2f11.1,i8,2i5,f7.1,f5.1,5f8.3,
868        1               i5,f9.3,3f7.4,5(1x,f9.4),i5,f12.7,f12.4)
869                 endif
870               endif
871  c
872               yearppt(ia) = 0.
873               yearrain(ia) = 0.
874               yearsnow(ia) = 0.
875               yearsncv(ia) = 0.
876               yearmelt(ia) = 0.
877               yearsubl(ia) = 0.
878               yeartet(ia) = 0.
879               yeardsw(ia) = 0.
880               yearnet(ia) = 0.
881               yearinf(ia) = 0.
882               yearoff(ia) = 0.
883               yearpet(ia) = 0.
884               yearbal(ia) = 0.
885               yearrun(ia) = 0.
886  c
887               aappt(ia) = 0.
888               aarain(ia) = 0.
889               aasnow(ia) = 0.
890               aasncv(ia) = 0.
891               aamelt(ia) = 0.
892               aasubl(ia) = 0.
893               aatet(ia) = 0.
894               aadsw(ia) = 0.
895               aapet(ia) = 0.
896               aanet(ia) = 0.
897               aainf(ia) = 0.
898               aaoff(ia) = 0.
899               aarun(ia) = 0.
900               aainfr(ia) = 0.
901  c
902  c
903   150   continue
904  C
905  C
906  c     ----------------------------------------------------
907  C==== SET DAILY ATMOSPHERIC PARAMETERS FOR YUCCA MOUNTAIN
908  ================
909  C.    ----------------------------------------------------
910        call atmos(ozone,wp,beta,csr,pg)
911  C
912  c     ---------------------------------
```

```
913   C==== FINISHED INITIALIZATION PROCEDURE
914   ==================================
915   c     ---------------------------------
916   c
917   c
918   c     Dayall = daily mass balance terms averaged for all points
919   c     (same as screen output)
920         open(unit=14,file=Dayall,status='unknown',recl=512)
921   c
922   c     write header info
923         write(14,161)
924   161   format(3x,'day',1x,'year',1x,'mo',1x,'dy',1x,'day',
925        1            2x,'temp',10x,'precip',11x,'rain',6x,'snow-fall',
926        2            5x,'snow-cover',6x,'snow-melt',4x,'sublimation',
927        3            4x,'pot-evaptrs',
928        3            4x,'evapo-trans',6x,'run-infil',7x,'del-soil',
929        4            6x,'net-infil',9x,'runoff',9x,'run-on',
930        5            6x,'fract-inf',3x,'it',2x,'discharge-#1',
931        6            4x,'discharge-#2',4x,'mass-balance',
932        7            4x,'mass-balance 2')
933   c
934   c
935   c     -------------------------
936   C==== BEGIN LOOPING THROUGH DAYS ==================================
937   c     -------------------------
938   c
939   c     J                  = simulation day number
940   c     ND                 = total simulation time in days
941   c     IA                 = location ID
942   c     MLOC          = total number of locations
943   c     NDAYS         = counter for spyglass output increment
944   c     ndays2        = counter for annual and mult-year output
945   c     ndays3        = number of days in year
946   C
947         ndays   = 0
948         ndays2  = 0
949         ndays3  = 0
950         ndays4  = 0
951         ndays5  = 0.
952         jold    = 0
953   c
954   c
955         do 4000 j = 1,nd
956            ndays2 = ndays2 + 1
957               ndays3 = ndays3 + 1
958   c
959   c           initialize runoff terms for each day
960               do 250 i = 1,nloc
961                 flowout(row(i),col(i)) = 0.
962                 flowin(locid(i))              = 0.
963                 totout(row(i),col(i))  = 0.
964                 totin(row(i),col(i))   = 0.
965   250         continue
966   C
967   c           ----------------------------------------
968   C======= LOOP THROUGH ALL LOCATIONS FOR EACH DAY
969   =======================
```

```
970   C            ----------------------------------------
971            do 3000 ia = 1,nloc
972   c
973                      tpet = 0.
974   c
975   c                  bypass outer boundary for flow routing logic (7/1/97)
976                      if(iwat(ia).eq.-3) goto 3000
977   c
978   C                  call subroutine for calculating solar radiation,
979   c                  net radiation, and potential evapotranspiration
980   c                  ----------------------------------------------------
981   -------
982                      call potevap (j,ia,dn,lat,lon,sl,asp,ridge,elev,
983         1                  dr,rd,pi,hstep,beta,pg,csr,wp,ozone,viewfactor,
984         2                  tr,tpet,row,col,atemp1,atemp2,avgairt,
985         3                  airtemp,iairtemp,trn)
986   c
987   c
988   c                  Call subroutine for spatially distributing daily
989   precip totals
990   c                  PPTLOC = average daily precip for each grid cell
991   c                  Use IPPTEST = 1 for mass balance testing
992   c                  mod 3/19/1999 aaprepx = average annual precip for
993   area
994   C                  of potential repository
995   c                  (used for scaling precip - elevation model)
996   c                  ----------------------------------------------------
997   -------
998                      call pptdist(ia,elev,pptfact,avgppt,pptyuc,pptfact2,
999         1                          row,col,pptloc,ppt,j,aaprepx)
1000                     if(ippttest.eq.1) pptloc(ia) = ppttest
1001  c
1002  c
1003  c                  Adust potential ET if precipitation occurs to account
1004  c                  for increased cloud cover and humidity (assumed
1005  function).
1006  c                  PETRS  = total daily potential evapotranspiration,
1007  c                  TPET   = advective term adjusted daily PET
1008  c                  Use IETEST = 1 to set PET to constant for testing
1009  c                  ----------------------------------------------------
1010  -------
1011                     petrs  = tpet/(((4*pptloc(ia))/25.4)+1)*etfact
1012  c
1013                     if(petrs.lt.0.) petrs = 0.
1014                     if(iettest.eq.1) petrs = ettest
1015                     petrs2 = petrs
1016                     tpet   = petrs*1.26
1017  c
1018  c                  added snow-cover module 3/29/99 (infils5d)
1019                     sublim(ia) = 0.
1020                     dsnowcv = 0.
1021                     melt = 0.
1022                     if(isnow.eq.2) then
1023                           call snow(ia,airtemp,snowmm,snopar1,
1024        1                        pptloc,rain,snowfall,melt,dsnowcv,
1025        2
1026         isublim,subpar1,sublim,tpet,isnwmod,trn,subpar2)
```

```
1027                    else
1028                          melt = 0.
1029                    endif
1030  c
1031  c                 mod 4/1/99 (infils5i)
1032  c                 if sublimation occurring, reduce trans for soil
1033  moisture
1034  c                 if airtemp < 0 deg C, turn off et from soil and
1035  bedrock
1036  c                 petrs2 = potential et term used by infil module
1037                    if(isnwmod.gt.0) then
1038                          if(sublim(ia).gt.0) petrs2 = petrs - sublim(ia)
1039                          if(petrs2.lt.0) petrs2 = 0.
1040                          if(airtemp.le.0) petrs2 = 0.
1041                    endif
1042  c
1043  c
1044  c---------- Call bucket for determining soil moisture,
1045  c                 et, net infil, and runoff
1046  c                 Use irout = -2 to bypass infiltration module for
1047  testing
1048  c                 added melt term 3/29/99 for snow-melt term (infils5d)
1049  c                 -----------------------------------------------------
1050  ---
1051                    if(irout.ne.-2) then
1052              call v4cbuck(j,dn,ia,soiltype,rocktype,ivegc,fvegc,
1053       1                  soilporo,fieldcap,imbibe,imb,soilks,sksfact,
1054       2                  soilalbeta,soiltalpha,soilresid,
1055       3                  pptloc,petrs,sorp,dsoilmm,tet,infilmm,
1056       4                  runoffmm,maxwgt1,maxwgt2,maxwgt3,maxwgt4,
1057       5                  massbal,finalvwc,soilmm,imbfact,
1058       6                  ibuck2,fdepth,rockpor,cdepth,rockmm,
1059       7                  rootf1,rootf2,rootf3,rootf4,idebug,
1060       8                  barsoil1,barsoil2,vegc,rocktalpha,rockalbeta,
1061       9
1062        rkmmfact,rkpor,infmod,etmod,rinfmm1,rinfmm2,melt,
1063       1                  snowmm,snowfall,rain,petrs2)
1064  c
1065                    else
1066                          runoffmm(ia) = pptloc(ia)
1067
1068                    endif
1069  c
1070  c---------- daily mass balance check for each location
1071              massbal = rain(ia)+snowfall(ia)-tet-dsoilmm-runoffmm(ia)
1072       1                     -infilmm(ia)-dsnowcv-sublim(ia)+rinfmm1(ia)
1073              locbal  = pptloc(ia)-rain(ia)-snowfall(ia)
1074  c
1075  c                 find maximum mass balance error for each location
1076              if(abs(massbal).gt.abs(maxbal(ia))) maxbal(ia) = massbal
1077  c
1078  c
1079  c---------- if runoff generated, set up flow routing terms
1080  c                 use irout = -1 to prevent flow routing
1081  c                 use irout = 0 to de-couple flow routing and
1082  infiltration
1083  c                 mod infils5o (5/12/99)
```

```
1084  c                 do not perform flow-routing on last day of simulation
1085                optmassb = 1
1086                if(((yr(j).eq.yrend).and.(dn(j).eq.dnend)).and.
1087        1             (optmassb.eq.1)) irout = -1
1088                if(irout.ne.-1) then
1089                      flowout(row(ia),col(ia)) = runoffmm(ia)
1090                else
1091                      flowout(row(ia),col(ia)) = 0.
1092                endif
1093                totflow     = totflow+flowout(row(ia),col(ia))
1094                rinfmm1(ia) = 0.
1095  c
1096  c
1097  c                 sum daily results for all locations
1098  c                 to calculate spatially averaged daily mass balance
1099  c                 and total daily mass balance
1100                      pptday   = pptday + pptloc(ia)
1101                      rainday  = rainday + rain(ia)
1102                      snowday  = snowday + snowfall(ia)
1103                      sncvday  = sncvday + snowmm(ia)
1104                      meltday  = meltday + melt
1105                      sublday  = sublday + sublim(ia)
1106                      atday    = atday + airtemp
1107                      petday   = petday + tpet
1108                      petrsday = petrsday + petrs
1109                      tetday   = tetday + tet
1110                      soilday  = soilday + dsoilmm
1111                      infday   = infday + infilmm(ia)
1112                      runday   = runday + runoffmm(ia)
1113                      dayvwc(ia)  = soilmm(1,ia)
1114  c
1115  c---------- calc annual totals for each location for map output
1116                      yearppt(ia)  = yearppt(ia) + pptloc(ia)
1117                      yearrain(ia) = yearrain(ia) + rain(ia)
1118                      yearsnow(ia) = yearsnow(ia) + snowfall(ia)
1119                      yearsncv(ia) = yearsncv(ia) + snowmm(ia)
1120                      yearmelt(ia) = yearmelt(ia) + melt
1121                      yearsubl(ia) = yearsubl(ia) + sublim(ia)
1122                      yearpet(ia)  = yearpet(ia) + tpet
1123  c                   yearpet(ia)  = yearpet(ia) + petrs
1124                      yeartet(ia)  = yeartet(ia) + tet
1125                      yeardsw(ia)  = yeardsw(ia) + dsoilmm
1126                      yearinf(ia)  = yearinf(ia) + infilmm(ia)
1127                      yearoff(ia)  = yearoff(ia) + runoffmm(ia)
1128                      yearbal(ia)  = yearbal(ia) + massbal
1129                      totbal       = totbal + yearbal(ia)
1130  c
1131  c
1132   3000         continue
1133  C
1134  c------- If runoff is generated then distribute runoff
1135  C            at end of day using surface flow routing
1136  C            ------------------------------------------------------------
1137                rinfday2 = rinfday
1138                rinfday = 0.
1139                infrday = 0.
1140                iter = 0
```

```
1141  c           if(totflow.gt.0.000000001) then
1142        if(totflow.gt.0.) then
1143              jifr = j
1144              call ifr(nloc,row,col,flowout,rinfmm1,rinfmm2,
1145       1                 soilporo,fieldcap,soiltype,soilmm,
1146       2                 soilvwc,buckmm,irout,imap,dn,j,
1147       3                 soilks,sksfact,rown,coln,totout2,totout,
1148       4                 iter,iwat,locid,nrun,cdepth,
1149       5                 imbibe,imbfact,rocktype,rockmm,
1150       6                 easting,northing,xcfs,ycfs,cfsday,idebug,
1151       7                 ifrtol,rkpor,rkmmfact,infilmm,flarea,
1152       8                 infmod,pptloc)
1153  c
1154  c
1155              do i = 1,nloc
1156                    yearrun(i)  = yearrun(i) + totout(row(i),col(i))
1157                    rinfday       = rinfday + rinfmm1(i)
1158                    infrday     = infrday + rinfmm2(i)
1159                    yearinfr(i) = yearinfr(i) + rinfmm2(i)
1160                    yearinfr(i) = yearinfr(i) + rinfmm1(i)
1161
1162              enddo
1163        endif
1164        totflow = 0.
1165  C
1166  c
1167  c------- calc annual totals and write output to map files
1168  c           ------------------------------------------------------------
1169  ---
1170        if((yr(j+1).gt.yr(j)).or.
1171       1      ((yr(j).eq.yrend).and.(dn(j).eq.dnend))) then
1172              ndays4 = ndays3
1173              ndays3 = 0
1174              nall   = nall + 1
1175  c
1176              if(tyear1.eq.0) then
1177                    noutyr = noutyr + 1
1178                    open(unit=15,file=outyear(noutyr),
1179       +                         status='unknown',recl=512)
1180  c
1181  c                       write header line
1182                    write(15,181)
1183  181               format(5x,'easting',3x,'northing',7x,'precip',
1184       1                 9x,'rain',5x,'snowfall',4x,'snowcover',
1185       2                 5x,'snowmelt',2x,'sublimation',4x,'evapotrns',
1186       3                 5x,'del-soil',4x,'net-infil',4x,'run-infil',
1187       4                 7x,'runoff',8x,'run-on',10x,'PET',
1188       5                 6x,'mass-balance')
1189  c
1190                        do 3011 i = 1,nloc
1191                              if(iwat(i).eq.-3) goto 3011
1192  c
1193                 write(15,180)  easting(i),northing(i),yearppt(i),
1194       1                        yearrain(i),yearsnow(i),yearsncv(i),
1195       2                        yearmelt(i),yearsubl(i),
1196       3                        yeartet(i),yeardsw(i),
1197       4                         yearinf(i),yearinfr(i),yearoff(i),
```

```
1198        5                          yearrun(i),yearpet(i),yearbal(i)
1199   180                 format(1x,2f11.1,1x,
1200        1                     7(f12.6,1x),4(f12.4,1x),
1201        2                     f14.4,1x,f12.4,2x,g16.9)
1202   c
1203                 aappt(i) = aappt(i) + yearppt(i)
1204          aarain(i) = aarain(i) + yearrain(i)
1205          aasnow(i) = aasnow(i) + yearsnow(i)
1206          aasncv(i) = aasncv(i) + yearsncv(i)
1207          aamelt(i) = aamelt(i) + yearmelt(i)
1208          aasubl(i) = aasubl(i) + yearsubl(i)
1209            aatet(i) = aatet(i) + yeartet(i)
1210            aadsw(i) = aadsw(i) + yeardsw(i)
1211          aapet(i) = aapet(i) + yearpet(i)
1212   c                            aapetrs(i) = aapetrs(i) + yrpetrs(i)
1213   c                       aanet(i) = aanet(i) + yearnet(i)
1214            aainf(i) = aainf(i) + yearinf(i)
1215            aaoff(i) = aaoff(i) + yearoff(i)
1216          aarun(i) = aarun(i) + yearrun(i)
1217          aainfr(i) = aainfr(i) + yearinfr(i)
1218   c                       totbal = totbal + yearbal(i)
1219   c
1220            yearppt(i) = 0.
1221          yearrain(i) = 0.
1222          yearsnow(i) = 0.
1223          yearsncv(i) = 0.
1224          yearmelt(i) = 0.
1225          yearsubl(i) = 0.
1226            yeartet(i) = 0.
1227          yearpet(i) = 0.
1228   c                            yrpetrs(i) = 0.
1229            yeardsw(i) = 0.
1230   c                       yearnet(i) = 0.
1231            yearinf(i) = 0.
1232            yearoff(i) = 0.
1233            yearbal(i) = 0.
1234            yearrun(i) = 0.
1235          yearinfr(i) = 0.
1236   c
1237   3011   continue
1238   c
1239          close(15)
1240          ndays2 = 0.
1241   c
1242   c
1243   c-------- calc annual averages for multi-year periods (using 365.25
1244   days/yr)
1245   c              write output to map files
1246   c              ------------------------------------------------------------
1247   ----
1248                       else if(((tyear1.eq.1).and.(nall.eq.tyear2)).or.
1249        1              ((yr(j).eq.yrend).and.(dn(j).eq.dnend))) then
1250                             noutyr = noutyr + 1
1251                             OPEN(UNIT=15,file=outyear(noutyr),
1252        +                                 status='unknown',recl=512)
1253   c
1254   c                         write header line
```

```
1255                          write(15,181)
1256  c181                    format(5x,'easting',3x,'northing',4x,'precip',
1257  c     1
1258        8x,'rain',4x,'snowfall',3x,'snowcover',
1259  c     2
1260        4x,'snowmelt',1x,'sublimation',3x,'evapotrns',
1261  c     3                             4x,'del-soil',3x,'net-
1262  infil',3x,'run-infil',
1263  c     4                             6x,'runoff',6x,'run-on',9x,'PET',
1264  c     5                             1x,'massbalance')
1265  c
1266                          ndays5 = float(ndays2)
1267                          do 3012 i = 1,nloc
1268                                if(iwat(i).eq.-3) goto 3012
1269  c
1270        aappt(i) = aappt(i) + yearppt(i)
1271        aarain(i) = aarain(i) + yearrain(i)
1272        aasnow(i) = aasnow(i) + yearsnow(i)
1273        aasncv(i) = aasncv(i) + yearsncv(i)
1274        aamelt(i) = aamelt(i) + yearmelt(i)
1275        aasubl(i) = aasubl(i) + yearsubl(i)
1276        aatet(i) = aatet(i) + yeartet(i)
1277  c                       aapetrs(i) = aapetrs(i) + yrpetrs(i)
1278        aadsw(i)  = aadsw(i) + yeardsw(i)
1279        aapet(i)  = aapet(i) + yearpet(i)
1280        aanet(i)  = aanet(i) + yearnet(i)
1281        aainf(i)  = aainf(i) + yearinf(i)
1282        aaoff(i)  = aaoff(i) + yearoff(i)
1283        aarun(i)  = aarun(i) + yearrun(i)
1284        aainfr(i) = aainfr(i) + yearinfr(i)
1285  c
1286        yearppt(i) = (yearppt(i)/ndays5)*365.25
1287        yearrain(i) = (yearrain(i)/ndays5)*365.25
1288        yearsnow(i) = (yearsnow(i)/ndays5)*365.25
1289        yearsncv(i) = (yearsncv(i)/ndays5)*365.25
1290        yearmelt(i) = (yearmelt(i)/ndays5)*365.25
1291        yearsubl(i) = (yearsubl(i)/ndays5)*365.25
1292        yearpet(i) = (yearpet(i)/ndays5)*365.25
1293        yeartet(i) = (yeartet(i)/ndays5)*365.25
1294        yeardsw(i) = (yeardsw(i)/ndays5)*365.25
1295        yearinf(i) = (yearinf(i)/ndays5)*365.25
1296        yearoff(i) = (yearoff(i)/ndays5)*365.25
1297        yearbal(i) = (yearbal(i)/ndays5)*365.25
1298        yearrun(i) = (yearrun(i)/ndays5)*365.25
1299        yearinfr(i) = (yearinfr(i)/ndays5)*365.25
1300  c                       totbal = totbal + yearbal(i)
1301  c
1302  c
1303           write(15,180)  easting(i),northing(i),yearppt(i),
1304        1                 yearrain(i),yearsnow(i),yearsncv(i),
1305        2                 yearmelt(i),yearsubl(i),
1306        3                 yeartet(i),yeardsw(i),
1307        4                 yearinf(i),yearinfr(i),yearoff(i),
1308        5                 yearrun(i),yearpet(i),yearbal(i)
1309  c     6                                                    dayvwc(i)
1310  c
1311  c                 write(15,180)  easting(i),northing(i),
```

```
1312  c     1                                yearppt(i),yeartet(i),yeardsw(i),
1313  c     2                                yearinf(i),yearinfr(i),yearoff(i),
1314  c     3
1315        yearrun(i),yearpet(i),dayvwc(i)
1316  c
1317  c
1318                              yearppt(i) = 0.
1319                                    yearrain(i) = 0.
1320                                    yearsnow(i) = 0.
1321                                    yearsncv(i) = 0.
1322                                    yearmelt(i) = 0.
1323                                    yearsubl(i) = 0.
1324                              yeartet(i) = 0.
1325                              yearpet(i) = 0.
1326  c                                 yrpetrs(i) = 0.
1327                              yeardsw(i) = 0.
1328                              yearnet(i) = 0.
1329                              yearinf(i) = 0.
1330                              yearoff(i) = 0.
1331                              yearbal(i) = 0.
1332                                    yearrun(i) = 0.
1333                                    yearinfr(i) = 0.
1334  c
1335  c
1336  3012                        continue
1337                              close(15)
1338                              nall   = 0
1339                              ndays5 = 0.
1340                              ndays2 = 0.
1341  c
1342                        endif
1343            endif
1344  C
1345  C------- calculate global average daily mass balance terms
1346               pptday   = pptday/float(ncell)
1347               rainday  = rainday/float(ncell)
1348               snowday  = snowday/float(ncell)
1349               sncvday  = sncvday/float(ncell)
1350               meltday  = meltday/float(ncell)
1351               sublday  = sublday/float(ncell)
1352  c
1353               atday    = atday/float(ncell)
1354               petday   = petday/float(ncell)
1355               tetday   = tetday/float(ncell)
1356               petrsday = petrsday/float(ncell)
1357               soilday  = soilday/float(ncell)
1358               infday   = infday/float(ncell)
1359               runday   = runday/float(ncell)
1360               totout3  = totout2/float(ncell)
1361               rinfday2    = rinfday2/float(ncell)
1362               infrday  = infrday/float(ncell)
1363  c
1364  c
1365  c------- calculate total daily discharge in cfs
1366  c            at user specified location (hard-wired for 30-meter grid)
1367  c            mod 3-21-1999 to use celsize from control file (infils5c)
1368  c
```

```
1369          totout2  = totout2/1000*celsize*celsize*(3.281**3.)/24/60/60
1370          cfsday   = cfsday/1000*celsize*celsize*(3.281**3.)/24/60/60
1371    c
1372    c
1373    c             calculate global average daily mass balance
1374          massbal2 = pptday+rinfday2-tetday-soilday-infday
1375         1            -runday-snowday+meltday
1376    c
1377                  daybal = pptday - rainday - snowday
1378    c
1379                  jold     = j
1380    c
1381    C****Comment out writes to unnecesary files for uncertainty analysis
1382    c
1383    c------- output global average daily mass balance terms
1384    c             mod to output formats 3/19/1999 (infils5b)
1385    C*            write(14,692) j,yr(j),month(j),day(j),dn(j),
1386    C*       1    atday,pptday,rainday,snowday,sncvday,meltday,
1387    C*       2    sublday,petrsday,
1388    C*       3    tetday,rinfday2,soilday,infday,runday,totout3,infrday,
1389    C*       4    iter,totout2,cfsday,massbal2,daybal
1390    c
1391    692           format(1x,i5,1x,i4,i3,i3,i4,1x,f5.1,1x,
1392         1                  14f15.9,1x,i3,3(1x,g15.9),3x,g15.9)
1393    c
1394    C*      write(*,693) yr(j),dn(j),pptday,atday,sncvday,petrsday,
1395    C*       1                tetday,rinfday2,soilday,infday,runday,totout3
1396    c
1397    c693          format(1x,i4,i4,f8.2,f5.0,2f6.2,2f8.2,3f8.2,1x,i2)
1398    693           format(1x,i4,i4,f6.1,f5.0,f6.1,2f5.1,2f7.1,3f7.1)
1399    c
1400    C*      write(13,694) j,yr(j),dn(j),pptday,atday,petday,petrsday,
1401    C*       1    tetday
1402    c
1403    694           format(1x,i5,i5,i4,f12.7,2x,f7.3,3f12.7)
1404    c
1405    c
1406    c------- sum global average daily mass balance for annual mass balance
1407    c             ------------------------------------------------------------
1408    --
1409                  pptyr  = pptyr + pptday
1410                  rainyr = rainyr + rainday
1411                  snowyr = snowyr + snowday
1412                  sncvyr = sncvyr + sncvday
1413                  meltyr = meltyr + meltday
1414                  sublyr = sublyr + sublday
1415    c
1416                  atyr    = atyr + atday
1417                  petyr   = petyr + petday
1418                  petrsyr = petrsyr + petrsday
1419                  tetyr   = tetyr + tetday
1420                  dswyr   = dswyr + soilday
1421                  infyr   = infyr + infday
1422                  rinfyr  = rinfyr + rinfday
1423                  runyr   = runyr + runday
1424                  outyr   = outyr + totout3
1425                  soutyr  = soutyr + totout3
```

```
1426  c
1427                pptday   = 0.
1428                rainday = 0.
1429                snowday = 0.
1430                sncvday = 0.
1431                meltday = 0.
1432                sublday = 0.
1433                atday    = 0.
1434                petday   = 0.
1435                petrsday = 0.
1436                tetday   = 0.
1437                soilday  = 0.
1438                infday   = 0.
1439  c             rinfday  = 0.
1440                runday   = 0.
1441                totout2  = 0.
1442                totout3  = 0.
1443                cfsday   = 0.
1444                iter     = 0
1445  c
1446            if((yr(j+1).gt.yr(j)).or.
1447        1   ((yr(j).eq.yrend).and.(dn(j).eq.dnend))) then
1448  c
1449  c                   atyr = atyr/365.25
1450  c                   write(9,305)
1451  yr(j),ppty r,atyr,petyr,petrsyr,tetyr,dswyr,
1452  c     1             infyr,runyr,outyr
1453  c305                format(1x,i4,9(1x,f9.3))
1454  c
1455                write(9,305) yr(j),ndays4,pptyr,rainyr,snowyr,sncvyr,
1456        1             meltyr,sublyr,
1457        2             petyr,petrsyr,tetyr,dswyr,
1458        3             infyr,runyr,outyr,totbal
1459  305                 format(1x,2i4,13(1x,f9.3),2x,g16.9)
1460  c
1461                      pptyr  = 0.
1462                      rainyr = 0.
1463                      snowyr = 0.
1464                      sncvyr = 0.
1465                      meltyr = 0.
1466                      sublyr = 0.
1467                      atyr = 0.
1468                      petyr = 0.
1469                      petrsyr = 0.
1470                      tetyr  = 0.
1471                      dswyr  = 0.
1472                      infyr  = 0.
1473                      runyr  = 0.
1474                      outyr  = 0.
1475                      totbal = 0.
1476  c
1477                endif
1478  c
1479  c
1480  c------- print map output for daily results
1481  c             ------------------------------------------------------------
1482        ndayout = ndaymap
```

```
1483          do i = 1,ndayout
1484                if((dn(j).eq.irday(i)).and.(yr(j).eq.nyrout(i))) then
1485                            open(unit=10,file=daymap(i),
1486        +                                status='unknown')
1487                do 1292 iii = 1,nloc
1488                if(iwat(iii).eq.-3) goto 1292
1489   c                  bckvwc = buckmm(iii)/depth(iii)/1000.
1490                      write(10,1291) easting(iii),northing(iii),j,
1491        1                 pptloc(iii),soilmm(1,iii),soilmm(2,iii),
1492        2                 soilmm(3,iii),rockmm(iii),
1493        2                 infilmm(iii),runoffmm(iii),
1494        2                 totout(row(iii),col(iii))
1495   c
1496   1292         continue
1497   1291         format(1x,f10.1,f11.1,1x,i5,7(1x,f10.4),1x,f16.4)
1498   c
1499                close(10)
1500                endif
1501                enddo
1502   C
1503   C
1504   C/////// end of day loop, go to next day ////////////////////////
1505   C=================================================================
1506   c
1507   3900  continue
1508   4000  continue
1509   4900  continue
1510   c
1511   c
1512   c---- calculate annual averages and write to ouput
1513   c     ---------------------------------------------
1514   c     flxfile = mapping file fo average annual mass balance terms
1515         open(unit=12,file=flxfile,status='unknown',recl=256)
1516   c
1517   c     write header line
1518         write(12,4915)
1519   4915  format(5x,'easting',3x,'northing',1x,7x,'precip',9x,'rain',
1520        1                 4x,'snow-fall',3x,'snow-cover',4x,'snow-melt',
1521        2                 2x,'sublimation',
1522        2                 3x,'evapotrans',4x,'run-infil',5x,'del-soil',
1523        3                 4x,'net-infil',7x,'runoff',11x,'run-on',
1524        4                 5x,'mass-balance',6x,'max-balance',
1525        5                 2x,'mass-balance #2')
1526   c
1527   c
1528   c     initialize summary statistics for global averages
1529         gaappt = 0.
1530         gaarain = 0.
1531         gaasnow = 0.
1532         gaasncv = 0.
1533         gaamelt = 0.
1534         gaasubl = 0.
1535         gaatet = 0.
1536         gaadsw = 0.
1537         gaapet = 0.
1538         gaanet = 0.
1539         gaainf = 0.
```

```
1540          gaaoff = 0.
1541          gaarun = 0.
1542          gaambl = 0.
1543          gaamb2 = 0.
1544    c
1545          fnd = float(nd)
1546          do 4910 i = 1,nloc
1547                if(iwat(i).eq.-3) goto 4910
1548    c
1549                aappt(i)  = aappt(i)*(365.25/fnd)
1550                aarain(i) = aarain(i)*(365.25/fnd)
1551                aasnow(i) = aasnow(i)*(365.25/fnd)
1552                aasncv(i) = aasncv(i)*(365.25/fnd)
1553                aamelt(i) = aamelt(i)*(365.25/fnd)
1554                aasubl(i) = aasubl(i)*(365.25/fnd)
1555                aatet(i)  = aatet(i)*(365.25/fnd)
1556                aadsw(i)  = aadsw(i)*(365.25/fnd)
1557                aapet(i)  = aapet(i)*(365.25/fnd)
1558    c           aapetrs(i) = aapetrs(i)*(365.25/nd)
1559                aanet(i)  = aanet(i)*(365.25/fnd)
1560                aainf(i)  = aainf(i)*(365.25/fnd)
1561                aaoff(i)  = aaoff(i)*(365.25/fnd)
1562                aarun(i)  = aarun(i)*(365.25/fnd)
1563                aainfr(i) = aainfr(i)*(365.25/fnd)
1564    c
1565    c           note: this mass balance check does not work
1566    c           if runoff is routed on last day (optmassb = 0)
1567                massb(i) = aarain(i) - aatet(i) - aadsw(i) - aainf(i)
1568          1                  -aaoff(i) + aainfr(i) + aamelt(i)
1569    c
1570                massb2(i) = aappt(i) - aarain(i) - aasnow(i)
1571
1572           write(12,4905) easting(i),northing(i),aappt(i),aarain(i),
1573          1                  aasnow(i),aasncv(i),aamelt(i),aasubl(i),
1574          2                  aatet(i),
1575          2
1576           aainfr(i),aadsw(i),aainf(i),aaoff(i),aarun(i),massb(i),
1577          3            maxbal(i),
1578          4        massb2(i)
1579    c
1580    4905          format(1x,2f11.1,1x,11f13.5,4(1x,g16.7))
1581    c
1582                gaappt = gaappt + aappt(i)
1583                gaarain = gaarain + aarain(i)
1584                gaasnow = gaasnow + aasnow(i)
1585                gaasncv = gaasncv + aasncv(i)
1586                gaamelt = gaamelt + aamelt(i)
1587                gaasubl = gaasubl + aasubl(i)
1588                gaatet = gaatet + aatet(i)
1589                gaadsw = gaadsw + aadsw(i)
1590                gaapet = gaapet + aapet(i)
1591    c           gaapetrs + geepetrs + aapetrs(i)
1592                gaanet = gaanet + aanet(i)
1593                gaainf = gaainf + aainf(i)
1594                gaaoff = gaaoff + aaoff(i)
1595                gaarun = gaarun + aarun(i)
1596                gaambl = gaambl + massb(i)
```

```
1597                gaamb2 = gaamb2 + maxbal(i)
1598   c
1599   4910  continue
1600         close(12,status='keep')
1601   c
1602   c
1603         soutyr = soutyr*(365.25/fnd)
1604   c
1605         gaappt = gaappt/float(ncell)
1606         gaarain = gaarain/float(ncell)
1607         gaasnow = gaasnow/float(ncell)
1608         gaasncv = gaasncv/float(ncell)
1609         gaamelt = gaamelt/float(ncell)
1610         gaasubl = gaasubl/float(ncell)
1611         gaatet = gaatet/float(ncell)
1612         gaadsw = gaadsw/float(ncell)
1613         gaapet = gaapet/float(ncell)
1614         gaanet = gaanet/float(ncell)
1615         gaainf = gaainf/float(ncell)
1616         gaaoff = gaaoff/float(ncell)
1617         gaarun = gaarun/float(ncell)
1618         gaamb1 = gaamb1/float(ncell)
1619         gaamb2 = gaamb2/float(ncell)
1620   c
1621         write(*,4955) gaappt,gaarain,gaasnow,gaasncv,gaamelt,gaasubl,
1622        1                  gaapet,gaatet,gaadsw,gaainf,gaaoff,gaarun,
1623        2             soutyr,gaamb1,gaamb2
1624         write(9,4955) gaappt,gaarain,gaasnow,gaasncv,gaamelt,gaasubl,
1625        1            gaapet,gaatet,gaadsw,gaainf,gaaoff,gaarun,
1626        2                    soutyr,gaamb1,gaamb2
1627   c
1628   4955  format(//1x,'Global Summary Statistics (mm/year):',
1629        1             /1x,'-----------------------------------------------
1630   --',
1631        2             //1x,'Precipitation..................... ',f16.6,
1632        3              /1x,'Rain............................ ',f16.6,
1633        4              /1x,'Snowfall........................ ',f16.6,
1634        5              /1x,'Snow-cover...................... ',f16.6,
1635        6              /1x,'Snow-melt....................... ',f16.6,
1636        7              /1x,'Sublimation..................... ',f16.6,
1637        7              /1x,'Potential Evapotranspiration.... ',f16.6,
1638        8              /1x,'Actual Evapotranspiration....... ',f16.6,
1639        9              /1x,'Change in Soil Moisture......... ',f16.6,
1640        1              /1x,'Net Infiltration................ ',f16.6,
1641        2              /1x,'Runoff Generation............... ',f16.6,
1642        3              /1x,'Cumulative Daily Run-on......... ',f16.6,
1643        4              /1x,'Outflow......................... ',f16.6,
1644        5              /1x,'Average Mass Balance Error...... ',g16.6,
1645        6        /1x,'Average Max Daily Error (mm/dy). ',g16.6,//)
1646
1647   c
1648         write(*,5) '  SIMULATION SUCCESSFULLY COMPLETED'
1649   c
1650   C
1651         close(11,status='keep')
1652         close(9,status='keep')
1653         close(13,status='keep')
```

```
1654          close(14,STATUS='keep')
1655          close(23,status='keep')
1656          if(ivwcflg.eq.1) close(16)
1657          stop
1658          end
1659    C
1660    C
1661    c===================== END OF MAIN PROGRAM
1662    ================================
1663    C/////////////////////////////////////////////////////////////////////
1664    /////
1665    C
1666    C/////////////////////////////////////////////////////////////////////
1667    /////
1668    C/////////////////    ATMOS SUBROUTINE
1669    ////////////////////////////////////
1670    C         ******** set atmospheric parameters, **********************
1671    c         ***** updated 12/11/95 by Don Burrows and Alan Flint ******
1672    C
1673    C        calulate clear sky solar radiation
1674    c
1675    C        OZONE -- ozone layer thickness (CM)
1676    c        WP    -- water in atmosphere(CM)
1677    C        BETA  -- mean atmospheric turbidity
1678    C        CSR   -- circumsolar radiation
1679    C        ELEV  -- meters elevation
1680    C        PG    -- reflectivity (surface)
1681    c
1682    C
1683          SUBROUTINE ATMOS(OZONE,WP,BETA,CSR,PG)
1684          DOUBLE PRECISION BETA(366),PG(366),WP(366),CSR(366),OZONE(366)
1685          DOUBLE PRECISION
1686    OZONE1,OZONE2,OZONE3,OZONE4,OZONE5,OZONE6,OZONE7,
1687         1        OZONE8,OZONE9,OZONE10,OZONE11,OZONE12,
1688         2        WP1,WP2,WP3,WP4,WP5,WP6,WP7,WP8,WP9,WP10,WP11,WP12,
1689         3        BETA1,BETA2,BETA3,BETA4,BETA5,BETA6,BETA7,
1690         4        BETA8,BETA9,BETA10,BETA11,BETA12,
1691         5        CSR1,CSR2,CSR3,CSR4,CSR5,CSR6,CSR7,CSR8,CSR9,
1692         6        CSR10,CSR11,CSR12,
1693         7        PG1,PG2,PG3,PG4,PG5,PG6,PG7,PG8,PG9,PG10,PG11,PG12
1694    C
1695          COMMON DBGFLAG
1696    c
1697          OZONE1=.29
1698          OZONE2=.31
1699          OZONE3=.32
1700          OZONE4=.33
1701          OZONE5=.33
1702          OZONE6=.32
1703          OZONE7=.30
1704          OZONE8=.29
1705          OZONE9=.28
1706          OZONE10=.27
1707          OZONE11=.27
1708          OZONE12=.28
1709          WP1=1.000
1710          WP2=1.000
```

```
1711          WP3=1.050
1712          WP4=1.100
1713          WP5=1.500
1714          WP6=1.800
1715          WP7=2.200
1716          WP8=2.440
1717          WP9=2.000
1718          WP10=1.400
1719          WP11=1.050
1720          WP12=0.950
1721          BETA1=.075
1722          BETA2=.075
1723          BETA3=.075
1724          BETA4=.085
1725          BETA5=.085
1726          BETA6=.090
1727          BETA7=.090
1728          BETA8=.084
1729          BETA9=.077
1730          BETA10=.075
1731          BETA11=.075
1732          BETA12=.075
1733          CSR1=.85
1734          CSR2=.85
1735          CSR3=.85
1736          CSR4=.85
1737          CSR5=.74
1738          CSR6=.74
1739          CSR7=.57
1740          CSR8=.57
1741          CSR9=.66
1742          CSR10=.74
1743          CSR11=.90
1744          CSR12=.90
1745          PG1=.24
1746          PG2=.24
1747          PG3=.24
1748          PG4=.24
1749          PG5=.24
1750          PG6=.24
1751          PG7=.24
1752          PG8=.24
1753          PG9=.24
1754          PG10=.24
1755          PG11=.24
1756          PG12=.24
1757    c
1758    c     set ozone and atmospheric turbidity terms as average for middle
1759    c     of each month (leep-year logic included as 1 extra day for
1760    December)
1761    c
1762          DO N=1,366
1763    C
1764    c     December
1765          IF (N .LE. 15.) THEN
1766             OZONE(N)=OZONE1 - ((OZONE1 - OZONE12) * ((15. - N) / 15.))
1767             WP(N) = WP1 - ((WP1 - WP12) * ((15. - N) / 15.))
```

```
1768            BETA(N) = BETA1 - ((BETA1 - BETA12) * ((15. - N) / 15.))
1769            CSR(N) = CSR1 - ((CSR1 - CSR12) * ((15. - N) / 15.))
1770            PG(N) = PG1 - ((PG1 - PG12) * ((15. - N) / 15.))
1771  C
1772        ELSE IF (N .GT. 349.) THEN
1773            OZONE(N) = OZONE1 - ((OZONE1 - OZONE12) * ((365. - N) / 15.))
1774            WP(N) = WP1 - ((WP1 - WP12) * ((365. - N) / 15.))
1775            BETA(N) = BETA1 - ((BETA1 - BETA12) * ((365. - N) / 15.))
1776            CSR(N)= CSR1 - ((CSR1 - CSR12) * ((365. - N) / 15.))
1777            PG(N) = PG1 - ((PG1 - PG12) * ((365. - N) / 15.))
1778
1779  C
1780  c     January
1781        ELSE IF (N .GT. 15 .AND. N .LE. 46) THEN
1782            OZONE(N) = OZONE2 - ((OZONE2 - OZONE1) * ((46.- N) / 31))
1783            WP(N) = WP2 - ((WP2 - WP1) * ((46.- N) / 31))
1784            BETA(N) = BETA2 - ((BETA2 - BETA1) * ((46.- N) / 31))
1785            CSR(N)= CSR2 - ((CSR2 - CSR1) * ((46.- N) / 31))
1786            PG(N) = PG2 - ((PG2 - PG1) * ((46.- N) / 31))
1787  C
1788  c     February
1789        ELSE IF (N .GT. 46. .AND. N .LE. 74.) THEN
1790            OZONE(N) = OZONE3 - ((OZONE3 - OZONE2) * ((74. - N) / 28.))
1791            WP(N) = WP3 - ((WP3 - WP2) * ((74. - N) / 28.))
1792            BETA(N) = BETA3 - ((BETA3 - BETA2) * ((74. - N) / 28.))
1793            CSR(N)= CSR3 - ((CSR3 - CSR2) * ((74. - N) / 28.))
1794            PG(N) = PG3 - ((PG3 - PG2) * ((74. - N) / 28.))
1795  C
1796  c     March
1797        ELSE IF (N .GT. 74. .AND. N .LE. 105.) THEN
1798            OZONE(N) = OZONE4 - ((OZONE4 - OZONE3) * ((105. - N) / 31.))
1799            WP(N) = WP4 - ((WP4 - WP3) * ((105. - N) / 31.))
1800            BETA(N) = BETA4 - ((BETA4 - BETA3) * ((105. - N) / 31.))
1801            CSR(N)= CSR4 - ((CSR4 - CSR3) * ((105. - N) / 31.))
1802            PG(N) = PG4 - ((PG4 - PG3) * ((105. - N) / 31.))
1803  C
1804  c     April
1805        ELSE IF (N .GT. 105. .AND. N .LE. 135.) THEN
1806            OZONE(N) = OZONE5 - ((OZONE5 - OZONE4) * ((135. - N) / 30.))
1807            WP(N) = WP5 - ((WP5 - WP4) * ((135. - N) / 30.))
1808            BETA(N) = BETA5 - ((BETA5 - BETA4) * ((135. - N) / 30.))
1809            CSR(N)= CSR5 - ((CSR5 - CSR4) * ((135. - N) / 30.))
1810            PG(N) = PG5 - ((PG5 - PG4) * ((135. - N) / 30.))
1811  C
1812  c     May
1813        ELSE IF (N .GT. 135. .AND. N .LE. 166.) THEN
1814            OZONE(N) = OZONE6 - ((OZONE6 - OZONE5) * ((166. - N) / 31.))
1815            WP(N) = WP6 - ((WP6 - WP5) * ((166. - N) / 31.))
1816            BETA(N) = BETA6 - ((BETA6 - BETA5) * ((166. - N) / 31.))
1817            CSR(N)= CSR6 - ((CSR6 - CSR5) * ((166. - N) / 31.))
1818            PG(N) = PG6 - ((PG6 - PG5) * ((166. - N) / 31.))
1819  C
1820  c     June
1821        ELSE IF (N .GT. 166. .AND. N .LE. 196.) THEN
1822            OZONE(N) = OZONE7 - ((OZONE7 - OZONE6) * ((196. - N) / 30.))
1823            WP(N) = WP7 - ((WP7 - WP6) * ((196. - N) / 30.))
1824            BETA(N) = BETA7 - ((BETA7 - BETA6) * ((196. - N) / 30.))
```

```
1825            CSR(N)= CSR7 - ((CSR7 - CSR6) * ((196. - N) / 30.))
1826            PG(N) = PG7 - ((PG7 - PG6) * ((196. - N) / 30.))
1827  C
1828  c     July
1829        ELSE IF (N .GT. 196. .AND. N .LE. 227.) THEN
1830            OZONE(N) = OZONE8 - ((OZONE8 - OZONE7) * ((227. - N) / 31.))
1831            WP(N) = WP8 - ((WP8 - WP7) * ((227. - N) / 31.))
1832            BETA(N) = BETA8 - ((BETA8 - BETA7) * ((227. - N) / 31.))
1833            CSR(N)= CSR8 - ((CSR8 - CSR7) * ((227. - N) / 31.))
1834            PG(N) = PG8 - ((PG8 - PG7) * ((227. - N) / 31.))
1835  C
1836  c     August
1837        ELSE IF (N .GT. 227. .AND. N .LE. 258.) THEN
1838            OZONE(N) = OZONE9 - ((OZONE9 - OZONE8) * ((258. - N) / 31.))
1839            WP(N) = WP9 - ((WP9 - WP8) * ((258. - N) / 31.))
1840            BETA(N) = BETA9 - ((BETA9 - BETA8) * ((258. - N) / 31.))
1841            CSR(N)= CSR9 - ((CSR9 - CSR8) * ((258. - N) / 31.))
1842                 PG(N) = PG9 - ((PG9 - PG8) * ((259. - N) / 31.))
1843  C
1844  c     September
1845        ELSE IF (N .GT.258. .AND. N .LE. 288.) THEN
1846            OZONE(N) = OZONE10 - ((OZONE10 - OZONE9) * ((288. - N) /
1847  30.))
1848            WP(N) = WP10 - ((WP10 - WP9) * ((288. - N) / 30.))
1849            BETA(N) = BETA10 - ((BETA10 - BETA9) * ((288. - N) / 30.))
1850            CSR(N)= CSR10 - ((CSR10 - CSR9) * ((288. - N) / 30.))
1851            PG(N) = PG10 - ((PG10 - PG9) * ((288. - N) / 30.))
1852  C
1853  c     October
1854        ELSE IF (N .GT. 288. .AND. N .LE. 319.) THEN
1855            OZONE(N) = OZONE11 - ((OZONE11 - OZONE10) * ((319.- N) /
1856  31.))
1857            WP(N) = WP11 - ((WP11 - WP10) * ((319.      - N) / 31.))
1858            BETA(N) = BETA11 - ((BETA11 - BETA10) * ((319.- N) / 31.))
1859            CSR(N)= CSR11 - ((CSR11 - CSR10) * ((319. - N) / 31.))
1860            PG(N) = PG11 - ((PG11 - PG10) * ((319. - N) / 31.))
1861  C
1862  c     November
1863        ELSE IF (N .GT.319. .AND. N .LE. 349.) THEN
1864            OZONE(N) = OZONE12 - ((OZONE12 - OZONE11) * ((349.- N) /
1865  30.))
1866            WP(N) = WP12 - ((WP12 - WP11) * ((349. - N) / 30.))
1867            BETA(N) = BETA12 - ((BETA12 - BETA11) * ((349. - N) / 30.))
1868            CSR(N)= CSR12 - ((CSR12 - CSR11) * ((349. - N) / 30.))
1869            PG(N) = PG12 - ((PG12 - PG11) * ((349. - N) / 30.))
1870        ENDIF
1871        ENDDO
1872  C
1873        RETURN
1874        END
1875  C
1876  C=================================================================
1877  =====
1878  c/////////////////////////////////////////////////////////////////
1879  /////
1880  C/////////////////// POTEVAP SUBROUTINE
1881  ////////////////////////////////////
```

```
1882  C
1883  c     start solar radiation routine
1884  c     find position of sun for each hour for each day
1885  C
1886        SUBROUTINE POTEVAP  (J,IA,DN,LAT,LON,SL,ASP,RIDGE,ELEV,
1887       1                     DR,RD,PI,HSTEP,
1888       2                     BETA,PG,CSR,WP,OZONE,VIEWFACTOR,
1889       3                     TR,TPET,row,col,atempl,atemp2,avgairt,
1890       4                                        airtemp,iairtemp,trn)
1891  C
1892  C
1893        INTEGER DN(400000)
1894        double precision avgairt(400000)
1895        INTEGER J,IA,DBGFLAG
1896        integer row(100000),col(100000)
1897        DOUBLE PRECISION LAT(100000),LON(100000),LAT2(100000)
1898        real elev(0:800,0:800)
1899        real H1,H2,HSTEP
1900        double precision SL(100000),SL2(100000),ASP(100000)
1901        double precision RIDGE(100000,0:36),RIDGE2(100000,0:36),
1902       1                 SLOPE(100000,0:36),HORIZ(100000,0:36)
1903
1904        DOUBLE PRECISION BETA(366),PG(366),WP(366),CSR(366),OZONE(366)
1905        DOUBLE PRECISION TAU,DEC,ET,ALT,AZ,STD,
1906       1     HASS,HASR,CF,LST,T,MA,IDR,IDA,IDM,ID,IR,ISS,IB
1907  C
1908        DOUBLE PRECISION II,COSTHETA,THETA,
1909       1     HA,ISC,PO,P,WO,AMSP,W,U1,U3,EO,TRR,TRO,ABO,TRG,ABW,TRW,
1910       2     TRA,TRAA,TRAS,FC,PAP,INN,SUNAZ,RNHORIZ
1911  c
1912        double precision zenith,idhoriz,ibhoriz,ibslope,rnslope
1913        double precision YRAD,DRAD,RAD,RN,GH,SSG,ALPHA
1914        double precision DR,RD,PI,TR,VIEWF,TA,PET,VIEWFACTOR
1915  c     real atfact
1916        double precision atempl,atemp2,pi2,taa,airtemp
1917        double precision dnn2,newatemp,trn
1918        double precision tpet
1919  C
1920  c     write(*,*) ia,airtemp ok
1921        DR    = 0.0174533
1922        RD    = 57.29579
1923  c
1924  C     standard meridian
1925        STD=120
1926  C
1927  C     latitude in radians
1928        LAT2(ia) = LAT(ia) * DR
1929  c
1930  C     slope in degrees
1931        SL2(ia) = ATAN(.01 * SL(ia)) * RD
1932        YRAD=0.
1933  c
1934  c
1935  C     *****  input elevation data for the surrounding ridges each 10
1936  ******
1937  C     *****************   startiNg at 10 degrees    *****************
1938  C     Input ridge elevation data each 10 deg starting at 10 deg
```

```
1939  c
1940          VIEWFACTOR = 0.
1941          DO II=1,36
1942           RIDGE2(ia,ii) = 90. - RIDGE(ia,II)
1943           COSTHETA = COS(SL2(ia)*DR)*COS((RIDGE2(ia,ii))*DR)
1944         1            +SIN(SL2(ia)*DR)
1945         2            *SIN((RIDGE2(IA,II))*DR)*COS((II*10.-ASP(ia))*DR)
1946           THETA = -ATAN(COSTHETA / SQRT(-COSTHETA ** 2. + 1.)) + 90. * DR
1947           THETA = 90. - THETA * RD
1948           VIEWF = 90. - THETA
1949           IF (VIEWF .GE. 90.) THEN
1950            VIEWF = 90.
1951           ENDIF
1952           VIEWFACTOR = VIEWFACTOR + VIEWF
1953          END DO
1954          VIEWFACTOR = VIEWFACTOR / (36. * 90.)
1955          II = 0
1956  c
1957  c
1958  C       SOLAR ROUTINE
1959  C       This subroutine is written to calculate the location of the sun
1960  C
1961  C
1962  C  ALT    ALTITUDE OF THE SUN (RADIANS)
1963  C  AZ     AZIMUTH OF THE SUN (RADIANS)
1964  C  LAT    LATITUDE OF THE SITE  (DEGREES)
1965  C  LON    LONGITUDE OF THE SITE (DEGREES)
1966  C  STD    STANDARD MERIDIAN OF THE SITE (DEGREES)
1967  C  DR     DEGREES TO RADIANS (DEGREES/RADIAN)
1968  C  RD     RADIANS TO DEGREES (RADIANS/DEGREE)
1969  C  CF     CORRECTION FACTOR FOR THE EQUATION OF TIME AND LONGITUDE
1970  CORRECTION (HOURS)
1971  C  LST    LOCAL STANDARD TIME (HOURS)
1972  C  T      LOCAL APPARENT TIME (HOURS)
1973  C  H1     STARTING HOUR FOR THE SIMULATION (HOURS)
1974  C  H2     ENDING HOUR FOR THE SIMULATION (HOURS)
1975  C  HSTEP  TIME STEP OF THE SIMULATION (HOURS)
1976  C  DN     DAY NUMBER (DAY OF THE YEAR, UNITLESS)
1977  C  DNN1    DAY OF THE YEAR TO BEGIN THE SIMULATION (UNITLESS)
1978  C  DN2    DAY OF THE YEAR TO END THE SIMULATION   (UNITLESS)
1979  c
1980  C
1981  c       TA   = air temperature (Kelvin) as a function of Julian day
1982  number
1983  c       TAU  = day angle (radians)
1984  c       DEC  = solar declination (radians)
1985  c       ET   = equation of time (minutes)
1986  c       HASR = hour angle of the sun at sunrise (radians)
1987  c       HASS = hour angle of the sun at sunset (radians)
1988  c
1989  c---- mod on 4/8/98: air temp shift (atfact)
1990  c       mod on 06/21/98 to distribute air temp by elevation
1991  c       mod on 7/13/98:
1992  c       user option when IAIRTEMP set to 1
1993  c       original model if iairtemp ne 1
1994  c
1995  c---- mod 9/2/98 newatemp = scaling function for future climate
```

```
1996  c     air temp - elevation correlation (data from Thompson et al)
1997  c     mod 3/18/99 (Infils5b): iairtemp = 3 for reading average
1998  c     daily air temp (C) from climate input file
1999  c
2000        dnn2 = float(dn(j))
2001        if(iairtemp.eq.3) then
2002              ta = (1400.-elev(row(ia),col(ia)))/1000.*9.8+273.15
2003              ta = ta + avgairt(j)
2004  c
2005        else if(iairtemp.eq.-1) then
2006              ta = avgairt(j) + 273.15
2007
2008        else if(iairtemp.eq.2) then
2009              newatemp = 0.006114*elev(row(ia),col(ia))-14.1819+5.3166
2010              ta  = ((atemp1-atemp2*sin(dnn2/366.*2.*pi+1.3)))
2011       1              +(1524.-
2012  elev(row(ia),col(ia)))/1000.*9.8+273.15+newatemp
2013        else if(iairtemp.eq.1) then
2014              ta  = ((atemp1-atemp2*sin(dnn2/366.*2.*pi+1.3)))
2015       1      +(1524.-elev(row(ia),col(ia)))/1000.*9.8+273.15
2016        else
2017              ta  = (atemp1-atemp2*sin(dnn2/366.*2.*pi+1.3))+273.15
2018        endif
2019        airtemp = ta - 273.15
2020  c
2021        TAU = 2*PI*(DN(J)-1)/365
2022  c
2023        DEC = (0.006918-0.399912*COS(TAU)+0.07257*SIN(TAU)
2024       1      -0.006758*COS(2*TAU)+0.000907*SIN(2*TAU)
2025       2            -0.002697*COS(3*TAU)
2026       3      +0.00148*SIN(3*TAU))
2027  c
2028        ET  = (0.000075+0.001868*COS(TAU)-0.032077*SIN(TAU)
2029       1      -0.014615*COS(2*TAU)-0.04089*SIN(2*TAU))*229.18
2030
2031        HASR = DACOS(-TAN(LAT2(ia))*TAN(DEC))
2032        HASS = -HASR
2033        H1   = 12.-HASR*RD/15.
2034        H2   = 12.-HASS*RD/15.
2035  c
2036  C     CORRECTION FACTOR FOR STANDARD TIME AT LOCAL LONGITUDE IN HOURS
2037        CF=(4*(STD-LON(ia))+ET)/60.
2038  C
2039  C----- LOOP THROUGH DAYLIGHT HOURS
2040  C      HOUR OF THE DAY LOOP MODEL RUNS FROM SUNRISE TO SUNSET
2041  c     trn = total net radiation for day (mod infils5i)
2042  C
2043        trn = 0.
2044        DO 2000 LST=H1,H2,HSTEP
2045  c
2046  C        CORRECTION FOR LOCAL TIME
2047           T  = LST+CF
2048           HA = 15.*(T-12.)*DR
2049  C
2050  C        ALTITUDE OF THE SUN AT HOUR ANGLE HA
2051           ALT = DASIN(SIN(DEC)*SIN(LAT2(ia))+COS(DEC)
2052       1          *COS(LAT2(ia))*COS(HA))
```

```
2053   c
2054   C         AZIMUTH OF THE SUN AT HOUR ANGLE HA
2055             if((SIN(ALT)*SIN(LAT2(ia))-SIN(DEC))/(COS(ALT)
2056        1         *COS(LAT2(ia))).lt.1.) then
2057             AZ = DACOS((SIN(ALT)*SIN(LAT2(ia))-SIN(DEC))/(COS(ALT)
2058        1         *COS(LAT2(ia))))
2059             else
2060             AZ = DACOS(1-((SIN(ALT)*SIN(LAT2(ia))-SIN(DEC))/(COS(ALT)
2061        1         *COS(LAT2(ia)))-1))
2062             endif
2063             IF (HA .LT. 0.) THEN
2064                AZ = PI-AZ
2065             ELSE IF (HA .GE. 0.) THEN
2066                AZ = AZ+PI
2067             END IF
2068
2069             ALT = ALT*RD
2070             AZ  = AZ*RD
2071
2072             IF (ALT .LE. 0.) THEN
2073                ZENITH = 90
2074             ELSE IF (ALT .GT.0.) THEN
2075                ZENITH = 90-ALT
2076             ENDIF
2077
2078   C         END OF SOLAR POSTION ROUTINE
2079   C
2080   C
2081   C------- begin routine for calculating net radiation ------------
2082   c
2083   c            ISC = solar constant
2084   c            PO  = pressure at sea level
2085   c            P   = pressure at site
2086   c            WO  = single scatter albedo by aerosols
2087   c
2088             ALPHA = 1.
2089             ISC   = 1367.
2090             PO    = 1013.25
2091             P     = EXP(-.0001184 * ELEV(row(ia),col(ia))) * PO
2092             WO    = .9
2093   c
2094   C------- set coefficients
2095             AMSP = (COS(ZENITH*DR)+.15 * (93.885-ZENITH)
2096        1       **(-1.253))**(-1.)
2097             MA   = AMSP * (P / PO)
2098             P    = EXP(-.0001184 * 0) * PO
2099   c
2100   C------- pressure at beta measurement
2101   c
2102             W  = WP(DN(J)) * (P / PO)**.75 * (273. / TA)**.5
2103             U1 = W * AMSP
2104             U3 = OZONE(DN(J)) * AMSP
2105   c
2106   c------- corrected floating point error 4/15/98
2107   c          EO = 1. + .033 * COS((2 * 3.14159 * DN(J) / 365.))
2108             EO = 1. + .033 * COS((2 * 3.14159 * dnn2 / 365.))
2109   c
```

```
2110  C------- eccentricity correction
2111            TRR  = EXP(-.0903 * MA**.84 * (1. + MA - MA**1.01))
2112            ABO  = ((.1611 * U3 * ((1 + 139.48 * U3)**(-.3035)))
2113       1          - .002715 * U3 * ((1.+.044*U3+.0003*(U3**2))**(-1)))
2114            TRO  = 1. - ABO
2115            TRG  = EXP(-.0127 * MA**.26)
2116            ABW  = 2.4959 * U1 *
2117       1          ((1.+79.034*U1)**.6828 + 6.385 * U1)**(-1.)
2118            TRW  = 1. - ABW
2119            TRA  = (.12445 * ALPHA - .0162) + (1.003 - .125 * ALPHA)
2120       1          * EXP(-BETA(DN(J)) * ALPHA * MA * (1.089 + .5120001))
2121            TRAA = 1. - (1. - WO) * (1. - MA + MA**1.06) * (1. - TRA)
2122            IDR  = EO*.79*ISC*COS(ZENITH*DR)*TRO*TRG*TRW*TRAA
2123       1          * .5 * (1. - TRR) * (1. - MA + MA**1.02) ** (-1.)
2124            TRAS = TRA / TRAA
2125            FC   = .84
2126            IDA  = EO*.79*ISC*COS(ZENITH*DR)*TRO*TRG*TRW*TRAA
2127       1          *FC*(1.-TRAS)/(1.-MA+MA**1.02)
2128            PAP  = .0685 + (1. - FC) * (1. - TRAS)
2129  c
2130  C------- direct beam (normal to surface)
2131            INN = EO * .9751 * ISC * TRR * TRO * TRG * TRW * TRA
2132  c
2133  C------- direct beam (horizontal surface)
2134            IB = INN * COS(ZENITH * DR)
2135  c
2136  C------- diffuse (multiple reflected)
2137            IDM = (IB+IDA+IDR) * (PG(DN(J)) * PAP / (1.-PG(DN(J))*PAP))
2138  c
2139  C------- block all radiation
2140            IF (ZENITH .GE. 90.) THEN
2141               IB = 0.
2142               INN = 0.
2143               IDR = 0.
2144               IDA = 0.
2145               IDM = 0.
2146            ENDIF
2147            SUNAZ = INT(AZ / 10. + .5)
2148  c
2149  C------- block by east-west ridge
2150            IF (RIDGE2(ia,SUNAZ) .LE. ZENITH) THEN
2151               IB = 0
2152            ENDIF
2153            IF ((RIDGE2(ia,SUNAZ) - 4.) .LE. ZENITH) THEN
2154               IDA = (1. - CSR(DN(J))) * IDA
2155            ENDIF
2156  c
2157  C------- diffuse radiation on a horizontal surface
2158            ID = IDR + IDA + IDM
2159  c
2160  C------- calculate solar angle of incidence, (90-THETA)
2161            COSTHETA = COS(SL2(ia)*DR) * COS(ZENITH*DR) + SIN(SL2(ia)*DR)
2162       1                * SIN(ZENITH*DR) * COS((AZ-ASP(ia))*DR)
2163            THETA    = -ATAN(COSTHETA / SQRT(-COSTHETA**2.+1.)) + 90. * DR
2164            THETA    = THETA * RD
2165            IF (THETA .GT. 90.) THEN
2166               THETA = 90.
```

```
2167            ENDIF
2168   c
2169   C------- add for ground reflection and subtract blocked sky
2170            IR = (ID + IB) * PG(DN(J)) * (1. - VIEWFACTOR)
2171            ISS = ID * VIEWFACTOR
2172   c
2173   C------- DIFFUSE RADIATION, SLOPE OR HORIZONTAL
2174            IDHORIZ = ISS + IR
2175            IDSLOPE = ISS + IR
2176   c
2177   C------- DIRECT BEAM RADIATION, SLOPE OR HORIZONTAL
2178            IBHORIZ = IB
2179            IBSLOPE = IBHORIZ * (COS(THETA * DR) / COS(ZENITH * DR))
2180            RNHORIZ = (IBHORIZ + IDHORIZ)
2181            RNSLOPE = (IBSLOPE + IDSLOPE)
2182            IF (RNHORIZ .LT. 0.) THEN
2183               RNHORIZ = 0.
2184            ENDIF
2185            IF (RNSLOPE .LT. 0.) THEN
2186               RNSLOPE = 0.
2187            ENDIF
2188            HORIZ(ia,II) = IBHORIZ + IDHORIZ
2189            SLOPE(ia,II) = IBSLOPE + IDSLOPE
2190   c
2191   C------- RADIATION (RAD) IS IN J/m^2 PER TIME STEP
2192            RAD = RNSLOPE*HSTEP*3600.
2193            TR  = TR + RAD
2194   c
2195   C------- CALCULATE NET RADIATIION
2196            RN = (-71*hstep*3600.)+.72*RAD
2197               trn = trn + rn
2198   c
2199   C------- CALCULATE SOIL HEAT FLUX
2200            GH   =  (-20*hstep*3600)+.386*RN
2201            SSG  =  -13.281+.083864*TA-.00012375*TA**2
2202            PET  =  SSG*(RN-GH)/2.45E6
2203            TPET =  TPET + PET
2204   c     write(*,*) tpet,et
2205   c
2206   C
2207   2000  CONTINUE
2208   C
2209   C---- END OF HOUR LOOP
2210   C
2211   c---- DRAD = total daily radiation ---------------------------
2212   C
2213         DRAD=TR
2214         TR = 0.
2215   C
2216         RETURN
2217         END
2218   c
2219   C=====================================================================
2220   ===========
2221   c//////////////////////////////////////////////////////////////////////
2222   ///////////
2223   c/////////////////// Cascading Bucket subroutine ////////////
```

```
2224  c
2225  c     mod to use snow-melt term 3/29/99 (infils5d)
2226  c     mod to use variable 4-layer cascading bucket routine
2227  c     mod to include dynamic root zone weighting
2228  c     (3 soil layers and 1 bedrock layer) (7/23/98)
2229  c
2230  c
2231        subroutine v4cbuck(j,dn,ia,soiltype,rocktype,ivegc,fvegc,
2232       1                   soilporo,fieldcap,imbibe,imb,soilks,sksfact,
2233       2                   soilalbeta,soiltalpha,soilresid,
2234       3                   pptloc,petrs,sorp,dsoilmm,tet,infilmm,
2235       4                   runoffmm,maxwgt1,maxwgt2,maxwgt3,maxwgt4,
2236       5                   massbal,finalvwc,soilmm,imbfact,
2237       6                   ibuck2,fdepth,rockpor,cdepth,rockmm,
2238       7                   rootf1,rootf2,rootf3,rootf4,idebug,
2239       8                   barsoil1,barsoil2,vegc,rocktalpha,rockalbeta,
2240       9
2241        rkmmfact,rkpor,infmod,etmod,rinfmm1,rinfmm2,melt,
2242       1                   snowmm,snowfall,rain,petrs2)
2243  c
2244        integer ia,ibuck2,idebug,infmod,etmod,j,ivegc
2245        integer soiltype(100000),rocktype(100000),topoid(100000)
2246        integer dn(400000)
2247        double precision vegc(100000)
2248        double precision cdepth(4,100000)
2249        double precision soilporo(20),fieldcap(20),imbibe(0:500)
2250        double precision rocktalpha(0:500),rockalbeta(0:500)
2251        double precision soilalbeta(20),soiltalpha(20),soilks(20)
2252        double precision runoffmm(100000),infilmm(100000)
2253        double precision soilmm(3,100000)
2254        double precision sorp(100000),soilresid(100000)
2255        double precision runonmm(100000)
2256        double precision rockpor(100000),rockmm(100000)
2257  c
2258        double precision pptloc(100000)
2259        double precision rinfmm1(100000),rinfmm2(100000)
2260        double precision snowmm(100000),snowfall(100000)
2261        double precision rain(100000)
2262  c
2263        double precision extra,tet,buckvwc,initvwc,finalvwc
2264        double precision fieldcp,pptvwc,spor,sresid
2265        double precision a,b,imbfact
2266        double precision fstore,fdepth,rootf1,rootf2,rootf3,rootf4
2267        double precision rpor,sks
2268        double precision fcmm1,fcmm2,fcmm3,rkmm
2269        double precision drmm1,drmm2,drmm3,drmm4,drmm5
2270        double precision pormm1,pormm2,pormm3
2271        double precision resmm1,resmm2,resmm3,resmm4
2272        double precision relsat1,relsat2,relsat3,relsat4,srelsat
2273        double precision et1,et2,et3,et4
2274        double precision soilmm1,soilmm2,soilmm3,rockmm1
2275        double precision vwc1,vwc2,vwc3,vwc4
2276        double precision massbal,dsoilmm,imb,initmm,finalmm
2277        double precision wgt1,wgt2,wgt3,wgt4
2278        double precision barsoil1,barsoil2,a2,b2
2279        double precision rkmmfact,etbackmm
2280        double precision rkpor,pptin,pptout
```

```
2281          double precision skmm,soilks1,sksfact,imbp,skmmp,fvegc
2282          double precision maxwgt1,maxwgt2,maxwgt3,maxwgt4
2283          double precision melt
2284          double precision petrs,petrs2
2285    c
2286    c
2287    c     initialize drainage terms
2288          drmm1 = 0.
2289          drmm2 = 0.
2290          drmm3 = 0.
2291          drmm4 = 0.
2292    c
2293    c     initialize root zone weights and saturation terms
2294          wgt1 = 0.
2295          wgt2 = 0.
2296          wgt3 = 0.
2297          wgt4 = 0.
2298    c
2299          vwc1 = 0.
2300          vwc2 = 0.
2301          vwc3 = 0.
2302          vwc4 = 0.
2303    c
2304          relsat1 = 0.
2305          relsat2 = 0.
2306          relsat3 = 0.
2307          relsat4 = 0.
2308    c
2309    c     set rooting density terms
2310    c     bedrock root density is function of soil depth
2311          rden1 = rootf1
2312          rden2 = rootf2
2313          rden3 = rootf3
2314          rden4 = cdepth(4,ia)*rootf4
2315          if(rden4.gt.1.) rden4 = 1.
2316    c
2317    c     initialize parameters
2318          a        = soiltalpha(soiltype(ia))
2319          b        = soilalbeta(soiltype(ia))
2320          a3       = rocktalpha(rocktype(ia))
2321          b3       = rockalbeta(rocktype(ia))
2322          a2       = barsoil2
2323          b2       = barsoil1
2324          imb      = imbibe(rocktype(ia))*imbfact
2325    c
2326    c     convert ks (kg sec /M^3 ) to mm/day
2327    c     use 86400*9.8 for kgm/sec^2 to mm/day conversion
2328    c     mult. by user specified scaler SKSFACT
2329          skmm  = soilks(soiltype(ia))* 86400. * 9.8 * sksfact
2330          if(soiltype(ia).eq.8) then
2331            skmm = imbibe(rocktype(ia))*imbfact
2332          endif
2333    c
2334    c     set infiltration capacity using soil ks
2335    c     and estimated storm duration
2336    c     winter storm = 12 hours, summer storms = 2 hours
2337
```

```
2338          if((dn(j).gt.183).and.(dn(j).lt.274)) then
2339                skmmp   = skmm/12.
2340                imbp    = imb/12.
2341          else
2342                skmmp   = skmm/2.
2343                imbp    = imb/2.
2344          endif
2345    c
2346    c
2347          infilmm(ia)  = 0.
2348          runoffmm(ia) = 0.
2349    c
2350    c
2351    c---- set field capacity terms for 3 soil layers
2352    c     rkpor is user specified input variable
2353    c     which defines effective bedrock storage capacity
2354          fieldcp  = fieldcap(soiltype(ia))
2355          fcmm1    = fieldcp*cdepth(1,ia)*1000.
2356          fcmm2    = fieldcp*cdepth(2,ia)*1000.
2357          fcmm3    = fieldcp*cdepth(3,ia)*1000.
2358          rkmm     = rkpor*cdepth(4,ia)*1000.
2359    c
2360    c---- set storage limit terms for 3 soil layers
2361    c     storage limit for bedrock = rkmm
2362          spor     = soilporo(soiltype(ia))
2363          pormm1   = spor*cdepth(1,ia)*1000.
2364          pormm2   = spor*cdepth(2,ia)*1000.
2365          pormm3   = spor*cdepth(3,ia)*1000.
2366    c
2367    c
2368    c---- set residual storage terms for soil layers
2369    c     residual for bedrock = 0.
2370          sresid   = soilresid(soiltype(ia))
2371          resmm1      = sresid*cdepth(1,ia)*1000.
2372          resmm2      = sresid*cdepth(2,ia)*1000.
2373          resmm3      = sresid*cdepth(3,ia)*1000.
2374          resmm4   = 0.
2375    c
2376    c
2377    c---- calculate intial storage for mass
2378    c     balance calculations.  Subtract increase in
2379    c     soil and bedrock storage caused by infiltrated runoff
2380    c     (rinfmm1 = change in soil & bedrock storage during
2381    c     runoff routing)
2382    c
2383          initmm = soilmm(1,ia) + soilmm(2,ia) +
2384         1                  soilmm(3,ia) + rockmm(ia) - rinfmm1(ia)
2385    c
2386    c---- step 1: add precip to upper soil profile
2387    c     and start forward cascade based on field capacity
2388    c     and soil permeability.
2389    c     Added snowmelt term (melt) 3/29/99 (infils5d)
2390    c     soilmm(1,ia) = soilmm(1,ia) + pptloc(ia) + melt
2391          soilmm(1,ia) = soilmm(1,ia) + rain(ia) + melt
2392    c
2393    c     set 1st layer permeability limit using precip intensity (skmmp)
2394    c     (top layer only INFILS4D, top 2 layers INFILS4C)
```

```
2395          if(soilmm(1,ia).gt.fcmm1) then
2396                drmm1 = soilmm(1,ia)-fcmm1
2397                if(drmm1.gt.skmmp) then
2398                      drmm1 = skmmp
2399                      soilmm(1,ia) = soilmm(1,ia) - skmmp
2400                else
2401   c                  soilmm(1,ia) = soilmm(1,ia) - drmm1
2402                      soilmm(1,ia) = fcmm1
2403                endif
2404          endif
2405   c
2406          soilmm(2,ia) = soilmm(2,ia) + drmm1
2407          if(soilmm(2,ia).gt.fcmm2) then
2408                drmm2 = soilmm(2,ia)-fcmm2
2409                if(drmm2.gt.skmm) then
2410                      drmm2 = skmm
2411                      soilmm(2,ia) = soilmm(2,ia) - skmm
2412                else
2413   c                  soilmm(2,ia) = soilmm(2,ia) - drmm2
2414                      soilmm(2,ia) = fcmm2
2415                endif
2416          endif
2417   c
2418          soilmm(3,ia) = soilmm(3,ia) + drmm2
2419          if(soilmm(3,ia).gt.fcmm3) then
2420                drmm3 = soilmm(3,ia)-fcmm3
2421                if(drmm3.gt.skmm) then
2422                      drmm3 = skmm
2423                      soilmm(3,ia) = soilmm(3,ia) - skmm
2424                else
2425   c                  soilmm(3,ia) = soilmm(3,ia) - drmm3
2426                      soilmm(3,ia) = fcmm3
2427                endif
2428          endif
2429   c
2430   c---- set net infil to imb (effective bedrock permeability)
2431   c      when rockmm > rkmm (effective bedrock storage capacity)
2432   c      if very shallow soil then limit bedrock infil by precip intensity
2433
2434   c
2435          if((cdepth(1,ia).lt.0.05).and.(drmm3.gt.imbp)) then
2436                soilmm(3,ia) = soilmm(3,ia) + drmm3-imbp
2437                drmm3 = imbp
2438          endif
2439          rockmm(ia) = rockmm(ia) + drmm3
2440          drmm5 = 0.
2441   c
2442   c      if bedrock storage exceeded then drainage <= imb
2443          if(rockmm(ia).gt.rkmm) then
2444                drmm4 = rockmm(ia)-rkmm
2445                if(drmm4.gt.imb) then
2446                      drmm4 = imb
2447                      rockmm(ia) = rockmm(ia) - imb
2448                else
2449   c                  rockmm(ia) = rockmm(ia) - drmm4
2450                      rockmm(ia) = rkmm
2451                endif
```

```
2452  c     endif
2453  c
2454  c     mod 08/10/98 to allow slow drainage as a function of
2455  c     saturation when rockmm > 0 but < rkmm
2456  c     if infmod = 1 then allow slow drainage function
2457  c     if infmod = 2 then allow full drainage: (drmm4 + drmm5 =< imb)
2458  c     note that slow drainage allowed only if cdepth4 > 0 (rkmm > 0.)
2459  c     if(rkmm.gt.0.) then
2460        else if(rockmm(ia).gt.0.and.rockmm(ia).le.rkmm) then
2461              if(rkmm.gt.0.) then
2462  c     if(infmod.eq.0) then
2463  c           drmm5 = 0.
2464  c
2465                    if(infmod.eq.1) then
2466  c                 if(rkmm.gt.0.) then
2467                          drmm5 = ((rockmm(ia)/rkmm)**rkmmfact)*imb
2468                          if(drmm5.gt.(imb-drmm4)) drmm5 = imb-drmm4
2469                          if(drmm5.gt.rockmm(ia)) then
2470                                drmm5 = rockmm(ia)
2471                                rockmm(ia) = 0.
2472                          else
2473                                rockmm(ia) = rockmm(ia) - drmm5
2474                          endif
2475  c                       else
2476  c                       drmm5 = 0.
2477  c                       endif
2478  c
2479                    else if(infmod.eq.2) then
2480  c                 if(rockmm(ia).gt.0.) then
2481                          drmm5 = imb - drmm4
2482                          if(drmm5.gt.rockmm(ia)) then
2483                                drmm5 = rockmm(ia)
2484                                rockmm(ia) = 0.
2485                          else
2486                                rockmm(ia) = rockmm(ia) - drmm5
2487                          endif
2488  c                 endif
2489                    endif
2490              endif
2491        endif
2492  c
2493        drmm4 = drmm4 + drmm5
2494  c====
2495  c
2496  c==== STEP 2: do reverse cascade to get excess water into top layer
2497  c     before et and final infil volumes computed
2498  c
2499        if(rockmm(ia).gt.rkmm) then
2500              soilmm(3,ia) = soilmm(3,ia) + rockmm(ia) - rkmm
2501              rockmm(ia) = rkmm
2502        endif
2503        if(soilmm(3,ia).gt.pormm3) then
2504              soilmm(2,ia) = soilmm(2,ia) + soilmm(3,ia) - pormm3
2505              soilmm(3,ia) = pormm3
2506        endif
2507        if(soilmm(2,ia).gt.pormm2) then
2508              soilmm(1,ia) = soilmm(1,ia) + soilmm(2,ia) - pormm2
```

```
2509                soilmm(2,ia) = pormm2
2510          endif
2511   c====
2512   c
2513   c==== STEP 3: remove et volume before calculating final infil and
2514   c     runoff volume. Set soil moisture to porosity
2515   c     if runoff volume exists
2516   c
2517          soilmm1 = soilmm(1,ia)
2518          if(soilmm1.gt.pormm1) soilmm1 = pormm1
2519          soilmm2 = soilmm(2,ia)
2520          if(soilmm2.gt.pormm2) soilmm2 = pormm2
2521          soilmm3 = soilmm(3,ia)
2522          if(soilmm3.gt.pormm3) soilmm3 = pormm3
2523          rockmm1 = rockmm(ia)
2524          if(rockmm1.gt.rkmm) rockmm1 = rkmm
2525
2526   c---- convert terms to relative saturation for et calculations
2527          if(cdepth(1,ia).gt.0.) vwc1 = soilmm1/cdepth(1,ia)/1000.
2528          if(cdepth(2,ia).gt.0.) vwc2 = soilmm2/cdepth(2,ia)/1000.
2529          if(cdepth(3,ia).gt.0.) vwc3 = soilmm3/cdepth(3,ia)/1000.
2530          if(cdepth(4,ia).gt.0.) vwc4 = rockmm1/cdepth(4,ia)/1000.
2531   c
2532          relsat1 = (vwc1-sresid)/(spor-sresid)
2533          if(relsat1.lt.0) relsat1 = 0.
2534          relsat2 = (vwc2-sresid)/(spor-sresid)
2535          if(relsat2.lt.0) relsat2 = 0.
2536          relsat3 = (vwc3-sresid)/(spor-sresid)
2537          if(relsat3.lt.0) relsat3 = 0.
2538          relsat4 = vwc4/rkpor
2539          if(relsat4.lt.0) relsat4 = 0.
2540   c
2541   c
2542   c---- calculate normalized dynamic root zone weights
2543   c     multiplied by root density terms
2544          srelsat = relsat1 + relsat2 + relsat3 + relsat4
2545          if(srelsat.gt.0.) then
2546                wgt1 = (relsat1/srelsat)*rden1
2547                wgt2 = (relsat2/srelsat)*rden2
2548                wgt3 = (relsat3/srelsat)*rden3
2549                wgt4 = (relsat4/srelsat)*rden4
2550          else
2551                wgt1 = 0.
2552                wgt2 = 0.
2553                wgt3 = 0.
2554                wgt4 = 0.
2555          endif
2556   c
2557   c     normalize root density terms and set upper weight limits
2558          swgt = wgt1+wgt2+wgt3+wgt4
2559          if(swgt.gt.0.) then
2560                wgt1 = wgt1/swgt
2561                if(wgt1.gt.maxwgt1) wgt1 = maxwgt1
2562                wgt2 = wgt2/swgt
2563                if(wgt2.gt.maxwgt2) wgt2 = maxwgt2
2564                wgt3 = wgt3/swgt
2565                if(wgt3.gt.maxwgt3) wgt3 = maxwgt3
```

```
2566                wgt4 = wgt4/swgt
2567                if(wgt4.gt.maxwgt4) wgt4 = maxwgt4
2568          endif
2569    c
2570    c
2571    c---- calculate et using Flint function and
2572    c     root-zone weights, then take et from soil
2573    c     and rock storage terms. For layer 1, use
2574    c     veg cover term to include bare soil evap
2575    c
2576    c     et1 = ((1-vegc(ia))*(a2*(1-exp(b2*relsat1))) +
2577    c     1          (vegc(ia))*(a*(1-exp(b*relsat1))))
2578    c
2579    c     mod 08/18/98
2580    c     if ivegc = 1, then use veg cover map data
2581    c     if ivegc = 0, use constant veg cover = fvegc
2582          if(ivegc.eq.1) then
2583                et1 = (1-vegc(ia))*(a2*(1-exp(b2*relsat1))) +
2584         1                  vegc(ia)*(a*(1-exp(b*relsat1)))
2585          else
2586                et1 = (1-fvegc)*(a2*(1-exp(b2*relsat1))) +
2587         1                  fvegc*(a*(1-exp(b*relsat1)))
2588          endif
2589          et1 = et1*wgt1*petrs2
2590          if(et1.lt.0) et1 = 0.
2591          soilmm(1,ia) = soilmm(1,ia) - et1
2592          if(soilmm(1,ia).lt.resmm1) then
2593                et1 = et1 - (resmm1 - soilmm(1,ia))
2594                soilmm(1,ia) = resmm1
2595          endif
2596    c
2597          et2 = a*(1-exp(b*relsat2))
2598          et2 = et2*wgt2*petrs2
2599          if(et2.lt.0) et2 = 0.
2600          soilmm(2,ia) = soilmm(2,ia) - et2
2601          if(soilmm(2,ia).lt.resmm2) then
2602                et2 = et2 - (resmm2 - soilmm(2,ia))
2603                soilmm(2,ia) = resmm2
2604          endif
2605    c
2606          et3 = a*(1-exp(b*relsat3))
2607          et3 = et3*wgt3*petrs2
2608          if(et3.lt.0) et3 = 0.
2609          soilmm(3,ia) = soilmm(3,ia) - et3
2610          if(soilmm(3,ia).lt.resmm3) then
2611                et3 = et3 - (resmm3 - soilmm(3,ia))
2612                soilmm(3,ia) = resmm3
2613          endif
2614    c
2615          et4 = a3*(1-exp(b3*relsat4))
2616          et4 = et4*wgt4*petrs2
2617          if(et4.lt.0) et4 = 0.
2618          rockmm(ia) = rockmm(ia) - et4
2619          if(soilmm(4,ia).lt.resmm4) then
2620                et4 = et4 - (resmm4 - soilmm(4,ia))
2621                soilmm(4,ia) = resmm4
2622          endif
```

```
2623
2624          tet = et1+et2+et3+et4
2625    c====
2626    c
2627    c==== STEP 4: If rockmm < rkmm then put drmm4 back into rockmm
2628    c     before calculating final net infil.
2629    c     daily ET demand must be satisfied for net
2630    c     infiltration to occur at end of day)
2631    c
2632          etbackmm = 0.
2633          if (etmod.eq.1) then
2634               etbackmm = rkmm - rockmm(ia)
2635                if(etbackmm.lt.0.) etbackmm = 0.
2636                if(etbackmm.gt.drmm4) then
2637                      etbackmm = drmm4
2638                      drmm4 = 0.
2639                else
2640                      drmm4 = drmm4 - etbackmm
2641                endif
2642          endif
2643    c
2644    c     added rinfmm2 to include infiltration from runoff (8/18/98)
2645          rockmm(ia)  = rockmm(ia) + etbackmm
2646          infilmm(ia) = drmm4 + rinfmm2(ia)
2647
2648    c====
2649    c
2650    c==== STEP 5: calculate final runoff volume
2651    c     add runoff volume obtained if precip intensity exceeded
2652    c     saturated soil permeability (skmm)
2653    c
2654          if(soilmm(1,ia).gt.pormm1) then
2655                runoffmm(ia) = soilmm(1,ia) - pormm1
2656                soilmm(1,ia) = pormm1
2657          endif
2658    c
2659    c
2660          finalmm = soilmm(1,ia) + soilmm(2,ia)
2661          1                      + soilmm(3,ia) + rockmm(ia)
2662          dsoilmm = finalmm - initmm
2663    c
2664    c     massbal  = pptloc(ia)-tet-dsoilmm-runoffmm(ia)-infilmm(ia)
2665    c     massbal  = rain(ia)+melt-tet-dsoilmm
2666    c     1                      -runoffmm(ia)-infilmm(ia)
2667    c
2668    c
2669    c---- added debugging option to check mass balance 8/2/98
2670          if(idebug.eq.1) then
2671                write(*,5) ia,pptloc(ia)
2672    5           format(1x,i5,1x,'precip =              ',f15.8)
2673                write(*,10) ia,initmm
2674    10          format(1x,i5,1x,'initial soilmm =      ',f15.8)
2675                write(*,15) ia,tet
2676    15          format(1x,i5,1x,'total et =            ',f15.8)
2677                write(*,16) ia,drmm4
2678    16          format(1x,i5,1x,'drmm4 =               ',f15.8)
2679                write(*,17) ia,drmm5
```

```
2680  17          format(1x,i5,1x,'drmm5 =                    ',f15.8)
2681              write(*,18) ia,imb
2682  18          format(1x,i5,1x,'imb =                      ',f15.8)
2683              write(*,20) ia,infilmm(ia)
2684  20          format(1x,i5,1x,'net infiltration =         ',f15.8)
2685              write(*,22) ia,runoffmm(ia)
2686  22          format(1x,i5,1x,'runoff =                   ',f15.8)
2687              write(*,25) ia,finalmm
2688  25          format(1x,i5,1x,'final soilmm =             ',f15.8)
2689              write(*,30) ia,dsoilmm
2690  30          format(1x,i5,1x,'change soilmm =            ',f15.8)
2691              write(*,35) ia,massbal
2692  35          format(1x,i5,1x,'mass balance =             ',f15.8)
2693              pause
2694        endif
2695        return
2696        end
2697  c
2698  C=================================================================
2699  ===========
2700  c/////////////////////////////////////////////////////////////////
2701  ///////////
2702  c/////////////////// added ifr subroutine 6/27/97 ////////////
2703  c
2704  c     Surface water flow routing routine (coupled)
2705  c     routing parameters defined using programs CHNNET16 and WATSHD20
2706  c
2707  c
2708        subroutine ifr(nloc,row,col,flowout,rinfmm1,rinfmm2,
2709        1                  soilporo,fieldcap,soiltype,soilmm,
2710       2                   soilvwc,buckmm,irout,imap,dn,j,
2711       3                   soilks,sksfact,rown,coln,totout2,totout,
2712       4                   iter,iwat,locid,nrun,cdepth,
2713       5                   imbibe,imbfact,rocktype,rockmm,
2714       6                   easting,northing,xcfs,ycfs,cfsday,idebug,
2715       7                   ifrtol,rkpor,rkmmfact,infilmm,flarea,
2716       8                   runmod,pptloc)
2717  c
2718  c
2719        integer dayn,edge,N2,irout,imap,jifr,rown,coln,runmod
2720        integer iter
2721  c
2722        double precision imbfact,imbfact2
2723        double precision rkmmfact
2724        double precision delay,buckvwc,totvwc,imb
2725        double precision fieldmm,poromm,infmm,maxmm
2726        double precision runmm,runmm2,xcfs,ycfs
2727        integer hole(100000),dn(400000)
2728        integer idir(0:800,0:800),locid(100000)
2729        integer row(100000),col(100000),soiltype(100000)
2730        integer rocktype(100000)
2731  c
2732        double precision cdepth(4,100000)
2733        double precision rinfmm1(100000),rinfmm2(100000),initmm(100000)
2734        double precision soilporo(20),fieldcap(20),soilks(20)
2735  c
2736        double precision flowin(300000),run(300000)
```

```
2737          integer iwat(100000)
2738          double precision soilvwc(100000)
2739          double precision infilmm(100000),pptloc(100000)
2740          double precision flowout(0:800,0:800),totout(0:800,0:800)
2741          double precision soilmm(3,100000),rockmm(100000)
2742          double precision imbibe(0:500)
2743          double precision easting(100000),northing(100000)
2744    c
2745          double precision pormm1,pormm2,pormm3,totmm
2746          double precision drmm1,drmm2,drmm3,drmm4,drmm5
2747          double precision fcmm1,fcmm2,fcmm3,rkmm
2748          double precision cfsday,finalmm
2749          double precision stormm1,stormm2,stormm3,stormm
2750          double precision ifrtol,rkpor,flarea,sksfact,imbp,skmmp
2751          double precision totout2,totflow
2752    c
2753    c
2754    c
2755    c---- initialize parameters
2756          imbfact2 = 0.
2757          drmm4 = 0.
2758          drmm5 = 0.
2759          cfsday = 0.
2760          delay = 1.
2761          iopt = 0
2762          iter = 0
2763          totout2 = 0.
2764    c
2765    c
2766    400   nrun = 0
2767          iter = iter + 1
2768          edge = 1
2769          totflow = 0.
2770    c
2771    c---- loop through all locations to rout surface flow
2772    c     do not loop through watershed boundary cells (iwat(i) = -3)
2773    c
2774          do 250 i = 1,nloc
2775                if(iwat(i).eq.-3) goto 250
2776    c
2777    c           calculate initial soil and bedrock water content
2778    c           for mass balance checking
2779                if(iter.eq.1) then
2780                      initmm(i) = soilmm(1,i) + soilmm(2,i) + soilmm(3,i)
2781          1                                 + rockmm(i)
2782                endif
2783    c
2784                idir(row(i),col(i)) = 0
2785                if(iter.gt.1) then
2786    c
2787    c---------- if irout = 1, do coupled flow routing
2788                if(irout.ge.1) then
2789                            runmm2 = 0.
2790                            pormm1  =
2791    soilporo(soiltype(i))*cdepth(1,i)*1000.
2792                            pormm2  =
2793    soilporo(soiltype(i))*cdepth(2,i)*1000.
```

```
2794                            pormm3  =
2795  soilporo(soiltype(i))*cdepth(3,i)*1000.
2796                            rkmm    = rkpor*cdepth(4,i)*1000.
2797  c
2798                            totmm = pormm1+pormm2+pormm3+rkmm
2799                            if(initmm(i).gt.totmm) then
2800                                  write(*,*) initmm(i),totmm
2801                                  pause
2802                            endif
2803  c
2804                            fieldcp = fieldcap(soiltype(i))
2805                            fcmm1   = fieldcp*cdepth(1,i)*1000.
2806                            fcmm2   = fieldcp*cdepth(2,i)*1000.
2807                            fcmm3   = fieldcp*cdepth(3,i)*1000.
2808  c
2809  c                         initialize drainage terms
2810                            drmm1 = 0.
2811                            drmm2 = 0.
2812                            drmm3 = 0.
2813                            drmm4 = 0.
2814                            drmm5 = 0.
2815  c
2816  c                         set infiltration capacity using soil ks
2817  c                         and estimated storm duration
2818  c                         winter storm = 12 hours
2819  c                         summer storms = 2 hours
2820  c                         convert ks (kg sec /M^3 ) to mm/day
2821  c                         mult by flow area (flarea) for cell area
2822  affected
2823  c                         assume precip infil does not affect infil of
2824  runoff volume
2825  c                         use 86400*9.8 for kgm/sec^2 to mm/day
2826  conversion
2827  c
2828                            skmm =
2829  soilks(soiltype(i))*86400.*9.8*sksfact*flarea
2830                            imb  = imbibe(rocktype(i))*imbfact2*flarea
2831                            if((dn(j).gt.183).and.(dn(j).lt.274)) then
2832                                  skmmp  = skmm/12.
2833                                  imbp   = imb/12.
2834                            else
2835                                  skmmp  = skmm/2.
2836                                  imbp   = imb/2.
2837                            endif
2838  c
2839  c                         write(*,*) imb,imbfact,imbibe(rocktype(i))
2840  c                         pause
2841  c                         initmm =
2842  soilmm(1,i)+soilmm(2,i)+soilmm(3,i)+rockmm(i)
2843
2844                            stormm1  = pormm1 - soilmm(1,i)
2845                            stormm2     = pormm2 - soilmm(2,i)
2846                            stormm3  = pormm3 - soilmm(3,i)
2847                            stormm   = stormm1 + stormm2 + stormm3
2848  c
2849                            if(idebug.eq.2) then
```

```
2850                                    write(23,1005)
2851  i,easting(i),northing(i),iter,j,
2852        1
2853        fcmm1,fcmm2,fcmm3,rkmm,pormm1,pormm2,pormm3,initmm,
2854        2
2855        soilmm(1,i),soilmm(2,i),soilmm(3,i),rockmm(i),
2856        3                           flowout(row(i),col(i)),flowin(i)
2857  1005                              format(1x,i8,2f11.1,2i5,14f10.4)
2858                              endif
2859  c
2860  c------------- start forward cascade
2861                              soilmm(1,i) = soilmm(1,i) + flowin(locid(i))
2862                              if(soilmm(1,i).gt.fcmm1) then
2863                                    drmm1 = soilmm(1,i)-fcmm1
2864                                    if(drmm1.gt.skmmp) then
2865                                          drmm1 = skmmp
2866                                          soilmm(1,i) = soilmm(1,i) - skmmp
2867                                    else
2868                                          soilmm(1,i) = soilmm(1,i) - drmm1
2869                                    endif
2870                              endif
2871  c
2872                              soilmm(2,i) = soilmm(2,i) + drmm1
2873                              if(soilmm(2,i).gt.fcmm2) then
2874                                    drmm2 = soilmm(2,i) - fcmm2
2875                                    if(drmm2.gt.skmmp) then
2876                                          drmm2 = skmmp
2877                                          soilmm(2,i) = soilmm(2,i) - skmmp
2878                                    else
2879                                          soilmm(2,i) = soilmm(2,i) - drmm2
2880                                    endif
2881                              endif
2882  c
2883                              soilmm(3,i) = soilmm(3,i) + drmm2
2884                              if(soilmm(3,i).gt.fcmm3) then
2885                                    drmm3 = soilmm(3,i) - fcmm3
2886                                    if(drmm3.gt.skmm) then
2887                                          drmm3 = skmm
2888                                          soilmm(3,i) = soilmm(3,i) - skmm
2889                                    else
2890                                          soilmm(3,i) = soilmm(3,i) - drmm3
2891                                    endif
2892                              endif
2893  c
2894  c                           mod INFILS3V 08/21/98
2895  c                           if shallow soil, then limit infil into bedrock
2896  c                           using estimated storm intensity
2897                              if((drmm3.gt.imbp).and.(cdepth(2,i).lt.2.0))
2898  then
2899                                    soilmm(3,i) = soilmm(3,i) + drmm3 - imbp
2900                                    drmm3 = imbp
2901                              endif
2902                              rockmm(i) = rockmm(i) + drmm3
2903                              drmm5 = 0.
2904  c
2905  c------------- Following modifications (starting INFILS3P) are to
2906  c                           allow net infil to occur during runoff routing
```

```
2907  c                          set net infil to imb (effective bedrock
2908  permeability)
2909  c                          when rockmm > rkmm (effective bedrock storage
2910  capacity)
2911  c                          mod 08/10/98 to allow slow drainage as a
2912  function of
2913  c                          saturation when rockmm > 0 but < rkmm
2914  c                          if infmod = 2 then allow slow drainage function
2915  c                          if infmod = 3 then allow full drainage:
2916  c                          (drmm4 + drmm5 =< imb in all cases)
2917  c
2918  c                          RUNMOD = 0, net infil occurs only during infil
2919  module
2920                             if(runmod.eq.0) then
2921                                   goto 99
2922
2923  c                          RUNMOD = 1, allow simple net infil only when
2924  rkmm exceeded
2925                             else if((runmod.eq.1).and.
2926        1                    (rockmm(i).gt.rkmm)) then
2927                                   drmm4 = rockmm(i)-rkmm
2928                                   if(drmm4.gt.imb) then
2929                                         drmm4 = imb
2930                                         rockmm(i) = rockmm(i) - imb
2931                                   else
2932                                         rockmm(i) = rockmm(i) - drmm4
2933                                   endif
2934  c
2935  c                          RUNMOD = 2,3, allow net infil when rkmm > 0.
2936                             else if(runmod.gt.1) then
2937                                   if (rockmm(i).gt.rkmm) then
2938                                         drmm4 = rockmm(i)-rkmm
2939                                         if(drmm4.gt.imb) then
2940                                               drmm4 = imb
2941                                               rockmm(i) = rockmm(i) - imb
2942                                         else
2943                                               rockmm(i) = rockmm(i) - drmm4
2944                                         endif
2945  c
2946  c
2947                                   else if((rockmm(i).gt.0).and.
2948        1                          (rockmm(i).le.rkmm)) then
2949  c
2950  c                                      RUNMOD = 2, allow slow drainage
2951  when rkmm > 0
2952                                         if(runmod.eq.2) then
2953  c                                      if(rockmm(ia).gt.0.) then
2954                                               drmm5 =
2955  ((rockmm(i)/rkmm)**rkmmfact)*imb
2956                                               if(drmm5.gt.(imb-drmm4))
2957  drmm5=imb-drmm4
2958                                               if(drmm5.gt.rockmm(i)) then
2959                                                     drmm5 = rockmm(i)
2960                                                     rockmm(i) = 0.
2961                                               else
2962                                                     rockmm(i) = rockmm(i) -
2963  drmm5
```

```
2964                                                  endif
2965  c                                         endif
2966  c
2967  c                                         RUNMOD = 3, allow full drainage
2968  when rkmm > 0
2969                                            else if (runmod.eq.3) then
2970  c                                         if(rockmm(ia).gt.0.) then
2971                                                  drmm5 = imb - drmm4
2972                                                  if(drmm5.gt.rockmm(i)) then
2973                                                        drmm5 = rockmm(i)
2974                                                        rockmm(i) = 0.
2975                                                  else
2976                                                        rockmm(i) = rockmm(i) -
2977  drmm5
2978                                                  endif
2979                                            endif
2980                                      endif
2981  c
2982                              endif
2983  c
2984  c
2985  c                           mod INFILS3V  08/21/98
2986  c                           calculate net infiltration and limit
2987  c                           maximum combined net infil to permeability
2988                              drmm4 = drmm4 + drmm5
2989                              if((drmm4+infilmm(i)).gt.imb) then
2990                                    rockmm(i) = rockmm(i) + drmm4-imb
2991                                    drmm4 = imb
2992                              endif
2993  c
2994  c
2995  c---------------- start reverse cascade to determine runoff
2996  c
2997  99                          continue
2998                              if(rockmm(i).gt.rkmm) then
2999                                    soilmm(3,i) = soilmm(3,i)+rockmm(i)-rkmm
3000                                    rockmm(i) = rkmm
3001                              endif
3002                              if(soilmm(3,i).gt.pormm3) then
3003                                    soilmm(2,i) = soilmm(2,i)+soilmm(3,i)-
3004  pormm3
3005                                    soilmm(3,i) = pormm3
3006                              endif
3007                              if(soilmm(2,i).gt.pormm2) then
3008                                    soilmm(1,i) = soilmm(1,i)+soilmm(2,i)-
3009  pormm2
3010                                    soilmm(2,i) = pormm2
3011                              endif
3012                              if(soilmm(1,i).gt.pormm1) then
3013                                    runmm2 = soilmm(1,i) - pormm1
3014                                    soilmm(1,i) = pormm1
3015                              endif
3016  c
3017                              if(runmm2.gt.0.) then
3018                                    flowout(row(i),col(i)) = runmm2
3019                              else
3020                                    flowout(row(i),col(i)) = 0.
```

```
3021                            endif
3022  c
3023                            flowin(locid(i)) = 0.
3024                            totout(row(i),col(i)) = totout(row(i),col(i)) +
3025        1                                   flowout(row(i),col(i))
3026  c
3027  c
3028  c---------- if irout ne 1, do decoupled flow routing
3029  c             ----------------------------------------
3030                 else
3031                    flowout(row(i),col(i)) = flowin(locid(i))*delay
3032                    totout(row(i),col(i))  = totout(row(i),col(i)) +
3033        1                                      flowout(row(i),col(i))
3034                    flowin(locid(i))  = flowin(locid(i))*(1-delay)
3035
3036                    endif
3037              endif
3038  c
3039                 finalmm = soilmm(1,i)+soilmm(2,i)+soilmm(3,i)+rockmm(i)
3040                 rinfmm1(i) = finalmm - initmm(i)
3041                 rinfmm2(i) = drmm4
3042  c
3043  c----- module testing 08/18/98
3044  c      write(*,*) j,i,iter,' pptloc  = ',pptloc(i)
3045  c      write(*,*) j,i,iter,' totmm   = ',totmm
3046  c      write(*,*) j,i,iter,' initmm  = ',initmm(i)
3047  c      write(*,*) j,i,iter,' finalmm = ',finalmm
3048  c      write(*,*) j,i,iter,' flowin  = ',flowin(locid(i))
3049  c      write(*,*) j,i,iter,' flowout = ',flowout(row(i),col(i))
3050  c      write(*,*) j,i,iter,' rinfmm1 = ',rinfmm1(i)
3051  c      write(*,*) j,i,iter,' rinfmm2 = ',rinfmm2(i)
3052  c      pause
3053  c
3054  c           truninf = truninf + runinf(i)
3055  c
3056  c-------- flow routing -------------------
3057  c           ------------
3058                flowin(iwat(i)) =
3059        1             flowout(row(i),col(i)) + flowin(iwat(i))
3060                run(iwat(i)) = flowin(iwat(i))
3061  c
3062                totflow = totflow + flowout(row(i),col(i))
3063  c
3064                if(flowout(row(i),col(i)).gt.ifrtol) then
3065                      irun = 1
3066                else
3067                      irun = 0
3068                endif
3069                      nrun = nrun + irun
3070  c
3071  c-------      sum outflow for specified cell, then convert
3072  c             to discharge units in main program (cfs)
3073                if(easting(i).eq.xcfs.and.northing(i).eq.ycfs) then
3074                      cfsday = flowout(row(i),col(i)) + cfsday
3075                endif
3076  c
3077  c
```

```
3078  250   continue
3079  c
3080  c
3081  c------ calculate total outflow
3082  c
3083        totout2 = 0.
3084        do 550 i = 1,nloc
3085              if(iwat(i).ne.-3) goto 550
3086              flowout(row(i),col(i)) =
3087        1     flowout(row(i),col(i)) + flowin(locid(i))
3088              flowin(locid(i)) = 0.
3089            totout2 = totout2 + flowout(row(i),col(i))
3090  550   continue
3091  c
3092  c
3093  c---- continue iteration if exces ppt still exists
3094  c     stop if more than 10 iterations
3095  c
3096        if(iter.gt.10) then
3097  c           do 750 i = 1,nloc
3098  c                 if(iwat(i).eq.-3) goto 750
3099  c                 if(flowout(row(i),col(i)).gt.0.) then
3100  c                       write(*,*)
3101  nrun,flowin(iwat(i)),flowout(row(i),col(i))
3102  c                 endif
3103  c750        continue
3104  c           pause
3105              goto 500
3106        endif
3107        if(nrun.gt.0) goto 400
3108  c
3109  c
3110  500   continue
3111        return
3112        end
3113  c
3114  C=================================================================
3115  ==========
3116  c/////////////////////////////////////////////////////////////////
3117  ///////////
3118  c/////////////////// added pptdist subroutine 6/27/97 ////////////
3119  c
3120  C-------- The next 5 lines added on 7/22/96 to use the 4JA - Yucca Mt.
3121  c         Precip transfer equation used in the Bucket Model
3122  c            mod 3/19/99 (infils5b) aaprepx = average annual precip
3123  (aap)
3124  c            over area of potential repository
3125  c         170 = average annual precip at 1400 m at Yucca Mountain
3126  c         133 = average annual precip at 4JA
3127  c         avgppt = average annual precip for input file (mod 7/28/96)
3128  c
3129        subroutine pptdist(ia,elev,pptfact,avgppt,pptyuc,pptfact2,
3130      1                    row,col,pptloc,ppt,j,aaprepx)
3131        integer ia,pptyuc,j
3132        integer row(100000),col(100000)
3133        double precision pptfact,pptfact2,avgppt
3134        real elev(0:800,0:800)
```

```
3135          double precision pptloc(100000),ppt(40000)
3136          real aaprepx
3137    c
3138    c---- modified on 6/21/98 (for infils1h)
3139    c     mod 3/19/99 (infils5b) pptyuc = 5 for monsoon climate
3140    c     mod 9/1/98 (infils4i) pptyuc = 4 for future climates
3141    c     pptyuc = 3 for Yucca Mt (4JA) using constant scaler
3142    c     pptyuc = 2 for Yucca Mt (4JA) using variable scaler
3143    c     pptyuc = 1 for analog climate simulations (Area 12, etc)
3144    c     pptyuc = (0) for uniform precip distribution
3145    c     - values used for original precip model
3146    c
3147          if(pptyuc.eq.5) then
3148          pptfact2 = pptfact*((exp(0.0006458*elev(row(ia),col(ia))+4.317)
3149         1              /aaprepx)/2. + 0.5)
3150    c
3151          else if(pptyuc.eq.4) then
3152          pptfact2 = pptfact*((exp(0.0006458*elev(row(ia),col(ia))+4.317)
3153         1           /170.)/2. + 0.5)
3154    c
3155          else if(pptyuc.eq.3) then
3156          pptfact2 = pptfact*(exp(0.0006458*elev(row(ia),col(ia))+4.317)
3157         1           /170.)*(170./133.)
3158    c
3159          else if(pptyuc.eq.2) then
3160          pptfact2 = pptfact*(exp(0.0006458*elev(row(ia),col(ia))+4.317)
3161         1           /170.)*(170./avgppt)
3162    c
3163          else if(pptyuc.eq.1) then
3164          pptfact2 = pptfact*(exp(0.0006458*elev(row(ia),col(ia))+4.317)
3165         1              /170.)
3166    c
3167    c
3168    c---- original 1996 model (added back 7/13/98)
3169    c
3170          else if(pptyuc.eq.-3) then
3171             PPTFACT2 = PPTFACT*(ELEV(row(IA),col(ia))/1400.)
3172         1                  *(170./133.)
3173          else IF(PPTYUC.EQ.-2) THEN
3174             PPTFACT2 = PPTFACT*(ELEV(row(IA),col(ia))/1400.)
3175         1                  *(170./avgppt)
3176          ELSE IF (PPTYUC.EQ.-1) THEN
3177             PPTFACT2 = PPTFACT*(elev(row(ia),col(ia))/1400.)
3178    c
3179    c---- uniform precip distribution
3180    c
3181          else
3182                pptfact2 = pptfact
3183
3184          endif
3185    c
3186    c     calculate location precip using daily precip input and pptfact2
3187          pptloc(ia) = ppt(j)*pptfact2
3188    c
3189          return
3190          end
3191    c
```

```
3192  C=================================================================
3193  ==========
3194  c/////////////////////////////////////////////////////////////////
3195  ///////////
3196  c/////////////////// added snow subroutine 3/29/99  (infils5d)
3197  ////////////
3198  c
3199
3200        subroutine snow(ia,airtemp,snowmm,snopar1,pptloc,rain,
3201       1                  snowfall,melt,dsnowcv,isublim,subpar1,sublim,
3202       2                  tpet,isnwmod,trn,subpar2)
3203  c
3204        integer ia,isublim,isnwmod
3205        double precision snowmm(100000),pptloc(100000)
3206        double precision snowfall(100000),rain(100000)
3207        double precision sublim(100000)
3208        double precision airtemp,melt,snowbeg,dsnowcv
3209        double precision trn,tpet
3210        real snopar1,subpar1,subpar2
3211  c
3212        melt = 0.
3213        rain(ia) = 0.
3214        snowfall(ia) = 0.
3215        snowbeg = snowmm(ia)
3216        sublim(ia) = 0.
3217  c
3218  c---- set rain = precip if above freezing
3219  c     (this will later be moved to pptdist module)
3220        if(airtemp.gt.0) then
3221              rain(ia) = pptloc(ia)
3222              snowfall(ia) = 0.
3223  c
3224  c---- calculate amount of melting if snow-cover > 0 and airtemp > 0
3225  c     isnwmod = 2 only available model for now
3226              if(snowmm(ia).gt.0) then
3227                    if(isnwmod.eq.2) then
3228                          melt = snopar1*airtemp
3229                    else
3230                          melt = 0.
3231                    endif
3232                    if(melt.gt.snowmm(ia)) melt = snowmm(ia)
3233                    snowmm(ia) = snowmm(ia) - melt
3234              endif
3235  c
3236  c---- convert precip to snowfall if airtemp below freezing
3237  c     use simple version for testing
3238  c     (this will later be moved to pptdist module)
3239  c
3240        else
3241              if(pptloc(ia).gt.0) then
3242                    snowfall(ia) = pptloc(ia)
3243                    rain(ia) = 0.
3244                    snowmm(ia) = snowmm(ia) + snowfall(ia)
3245              endif
3246        endif
3247  c
3248  c
```

```
3249  c     mod 4/1/99 (infils5h)
3250  c---- calculate snow-cover sublimation rate for dry days,
3251  c     as a function of air temp (this function is preliminary)
3252  c     sublimation = 0. if precip occurring
3253  c     do not let sublimation rate exceed potential et rate
3254  c
3255        if((isublim.eq.1).and.(pptloc(ia).eq.0.)) then
3256              sublim(ia) = ((20. + airtemp)/10.) * subpar1
3257              if(sublim(ia).lt.0) sublim(ia) = 0.
3258              if(sublim(ia).gt.snowmm(ia)) sublim(ia) = snowmm(ia)
3259              if(sublim(ia).gt.tpet) sublim(ia) = tpet
3260              snowmm(ia) = snowmm(ia) - sublim(ia)
3261
3262  c     test model
3263        else if((isublim.eq.2).and.(pptloc(ia).eq.0.)) then
3264              sublim(ia) = trn/2.501
3265              if(sublim(ia).lt.0) sublim(ia) = 0.
3266              if(sublim(ia).gt.snowmm(ia)) sublim(ia) = snowmm(ia)
3267              if(sublim(ia).gt.tpet) sublim(ia) = tpet
3268              snowmm(ia) = snowmm(ia) - sublim(ia)
3269
3270        else if(isublim.eq.3) then
3271              if(airtemp.gt.0.) sublim(ia) = subpar2 * tpet
3272              if(airtemp.le.0.) sublim(ia) = subpar1 * tpet
3273              if(sublim(ia).gt.subpar2*tpet) sublim(ia) = subpar2 * tpet
3274              if(sublim(ia).gt.snowmm(ia)) sublim(ia) = snowmm(ia)
3275              snowmm(ia) = snowmm(ia) - sublim(ia)
3276
3277        endif
3278  c
3279        dsnowcv = snowmm(ia) - snowbeg
3280  c
3281  c---- debugging
3282  c     write(*,101) ia,airtemp,pptloc(ia),rain(ia),snowfall(ia),
3283  c     1                  snowmm(ia),melt
3284  c101  format(1x,i8,1x,f6.2,1x,5f7.2)
3285  c     pause
3286  c
3287        return
3288        end
3289  c
3290  c//////////////////////////////// end of program
3291  ////////////////////////////////////////
3292  c//////////////////////////////////////////////////////////////////////
3293  ///////////////
3294  c
3295  c
3296
```

ATTACHMENT III

Documentation of PREINFIL (to satisfy AP-SI.1Q)

This attachment and Attachment IV contain documentation necessary to satisfy procedure AP-SI.1Q, Section 5.1.1.2 verification of routines used. PREINFIL can be considered to be a routine since it can be verified by visual inspection and hand calculations. Its purpose is to create automatically and readily input user control files for the computer model INFIL as used in these analyses. Attachment IV contains a listing of the source code for PREINFIL.

PREINFIL Version A1.20 software routine verification showing conformance to AP-SI.IQ Revision 2 ICN 4, Section 5.1.1.2.

Creation of INFIL Version 2.a1 input control files containing the following:

1. sampled parameter data using LHS, and
2. designated filenames,
3. As extracted from Procedure AP-SI.IQ Revision 2 ICN 4, Section 5.1.1.2:

> **5.1.1.2 Software Developer:**
> Include, as a minimum the following information:
> - Identification, including version of the software routine or macro.
> - The name and version identification of the commercial software under which the routine or macro was developed.
> - Sufficient documentation that the software routine or macro provides correct results for a specified range of input parameters. This includes inputs; computer generated results for a specified range of input parameters; computer generated evidence of the programmed algorithms or equations, e.g., computer program listings and spreadsheet cell contents; and verification results.

Documentation to satisfy first and second bullets of Section 5.1.1.2:

A software routine (consisting of several modules) was developed in accordance with AP-SI.1Q , Section 5.1.1.2 for the purpose of readily and automatically creating input control files compatible with INFIL V A_2.a1 which also incorporate designated 1) sampled parameter values (using Latin Hypercube Sampling) and 2) filenames which are read from this file and subsequently output by INFIL. The software developed, PREINFIL, V. 1.20, is in FORTRAN 77. The source code, which includes a change history, and the executable reside in the following directory location on a DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories:

I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.PREINFIL.SOURCE_CODE .NEW].

This satisfies the requirements of Section 5.1.1.2, first bullet of AP-SI.1Q for routines.

Documentation to satisfy third bullet of Section 5.1.1.2:

Satisfaction of the second bullet of Section 5.1.1.2 is accomplished by utilizing a sample input user file produced by PREINFIL for the process of executing INFIL V A_2.a1, and requiring that 1) the INFIL code terminates normally, 2) the correct values as sampled by LHS V. 2.50 are placed correctly into the same input control files, 3) the files with designated filenames (as input in the same input user control files are output as such by INFIL, and 4) the output files produced are equivalent to output files produced using the same input by an independently designed calculation. PREINFIL requires the following input; 1) a (binary) 'computational' database file (CDB), 2) a user control file which contains instructions for PREINFIL, and 3) a template input control file for INFIL V A_2.a1. The output from PREINFIL consists of a new input control file for INFIL V A_2.a1, satisfying the 4 items listed above, and an optional debug file.

Verification of requirement 1) can be obtained by noting that INFIL terminates normally.

To verify requirement 2), the following is done. Examine the input CDB file using a software tool called GROPECDB to determine the values of sampled parameters. A sample ascii output printout of the CDB file using GROPECDB is shown in Table III-3. This file shows the values of the sampled parameters from LHS (the parameter called ATEMP1 in the CDB file was not sampled). The input template user control file for INFIL that is modified is found in Table VII-8 (Attachment VII). Table III-1 and III-2 shows the corresponding input user control file for INFIL Version A_2.a1 produced by PREINFIL and manually, respectively. A comparison of the Table III-1 and III-2 shows the bolded entries, rounded to three significant decimal digits, are equivalent to the bolded values in Table III-3. One exception is for the parameter ETCOEFFA in Table III-3, corresponding to BARSOIL1 in Table III-1 and III-2, has a sign change which is done within PREINFIL. To determine the correspondence between names of parameters in the INFIL input user control file and the names in the GROPECDB listing, refer to the Input User Control File for PREINFIL (Table VII-7 in Attachment VII).

Verification of requirement 3) can be obtained by noting that the names of output files produced by INFIL (bold italics in Table III-1) are identical to the names as specified in the input user control file for PREINFIL.

Verification of item 4) is contained in the following argument. The independently designed calculation is one in which the input user control file is produced by hand using the same template as PREINFIL, and substituting the sampled parameter values as found in the input CDB file. A comparison of the summary output files produced by the INFIL code using the input user control file produced by PREINFIL, see Table III-4, Summary Output File from Executing INFIL Using PREINFIL to produce an Input Control File (Table III-1), and by hand, see Table III-5, Summary Output File from Executing INFIL Using Manual Substitutions into a Template Input Control File (Table III-2) shows that they are equivalent calculations. Figure III-1 shows this diagrammatically. Note specifically the output parameter (Net Infiltration) of importance here (bolded) is identical in both Table III-4 and III-5.

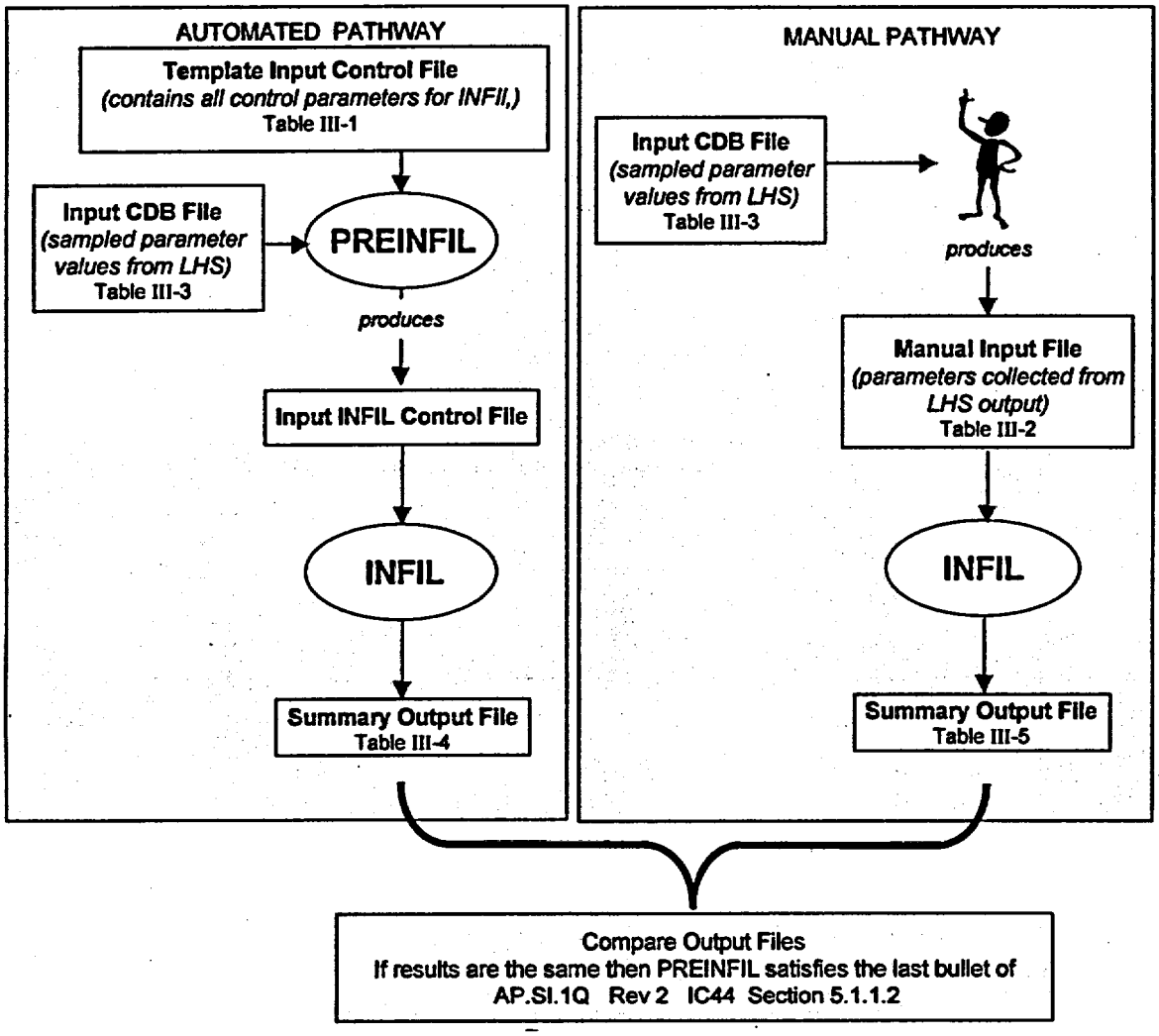


Figure III-1 Process of Verifying PREINFIL According to AP-SI.1Q

Table III-1. Sample Input User Control File for INFIL Version A_2.a1 produced by PREINFIL. (PREINFIL presents bolded values generally to three significant decimal digits, except for values corresponding to BARSOIL1, BARSOIL2, which have four)

```
INFIL2a1.ctl: Solitario East, Tule Lake, ww1-2a1-gm1-k2-w20 (11/18/1999)

1       0.00000001     IROUT(1 = coupled, 0 = uncoupled, -1 = flow routing off, -2 = infil off), IFRTOL
2       2        2.16E+00      ISNOW ISNWMOD SNOPAR1
3       1.57E-01       3.00E-01       ISUBLIM SUBPAR1 SUBPAR2
0       1.      IPPTTEST (1 for testing), PPTTEST = constant
0       .0      IETTEST (1 for testing), ETTEST = constant
3       0.0     CELSIZE (node spacing (meters): using for flow volume 31.0 calculations)
544721.0   4072203.0                          xcfs,ycfs: coords for discharge cell
1950   1   1996   366   1   5                 yr1=start year, dnn1=start day, yr2=end year, dn2=end day
1.12E+00   9.01E-01   5.48E+00  2.23E+00      MULTIPLI (PPTFACT ETFACT IMBFACT SKSFACT)
1.13E+00   0   4.00E-01                       SDFACT (SOIL DEPTH
1.0   0.8   0.8   0.2                         ROOTF1,ROOTF2,ROOTF3,ROOTF4
1.0   1.0   0.5   0.1                         MAXWGT1,MAXWGT2,MAXWGT3,MAXWGT4
0.30   2.00   3.75   2.00                     RDEPTH1 RDEPTH2 RDEPTH3 RDEPTH4
2.95E-02  1.00E+00  3.04E-01                  RKPOR RKMMFACT FLAREA
1   1   0                                     INFMOD, ETMOD, RUNMOD
-10.610   1.102                               BARSOIL1 BARSOIL2
3   17.3   11.74                              IAIRTEMP=1 for new air temp model, ATEMP1=avg. air temp,
       ATEMP2 = air temp seaso
2                                            HSTEP: time step for PET model (hours)
5   181   3                                  PPTYUC (=5 diminished elev. correlation, =2 for 4JA , =1 for
       simple elevation transf
Tulelake.inp                                 input file name:  daily precip
0      1    0.5      1     1          dpthflag, irtz, delvwcf, moistcr, fracmod
0      1.75                           IVWCFLG, vwcfact
ww1.w20                                      input file name:  map parameters (*.inp)
-1
ww1-gm1.2a1                                  dayall: average daily mass balance terms
1    1 ndaymap, imap
70   1995
ww1-gm1.2a2
INFILSM_WW1_R001.OUT
INFILMN_WW1_R001.OUT
1   1                                dbgflag, dbgflag2
ww1-gm1.2a5                                  output file name:  annual mass balance terms
0                                            IDEBUG: debugging option parameter
INFILECHO_R001.OUT
ww1-gm1                                      map output:  annual totals or mult-year averages
--
--------------- Parameters for dynamic root-zone function -----------------------------------
--     depth  rtza   rtzb   rtzc  rtzd  bsoil    delvwc
--     m
```

```
     4
     1    0.5     15      3     3     2     0.30      0.50
     2    1.5     15      4     2     2     0.20      0.50
     3    4.5     10    1.5     1     2     0.10      0.50
     4    6.0     10    1.5     1     2     0.05      0.25
--
--------------- Soil Properties (Brooks & Corey/van Genuchten combined) ----------------------------------------
--     fdcp   etresidpor    beta  alpah ksat     PE         B     n     vg-alpha sorp     SOILP   potis
--                                    Kg s /m^3J/Kg                    1/(J/Kg) ???              J/Kg
    10
     1  0.242  0.054  0.366  -3.5  1.26  5.6E-04 -1.19E+00  4.72  1.24  5.2E-01    0.390    0.05 -1.0E+02
     2  0.173  0.023  0.315  -3.5  1.26  1.2E-03 -9.41E-01  3.70  1.31  6.2E-01    0.500    0.05 -1.0E+02
     3  0.163  0.017  0.325  -3.5  1.26  1.3E-03 -8.60E-01  3.36  1.36  6.6E-01    0.510    0.05 -1.0E+02
     4  0.073  0.002  0.281  -3.5  1.26  3.8E-03 -6.22E-01  2.18  1.62  8.7E-01    0.700    0.05 -1.0E+02
     5  0.200  0.035  0.330  -3.5  1.26  6.7E-04 -1.07E+00  4.14  1.78  5.6E-01    0.400    0.05 -1.0E+02
     6  0.150  0.011  0.339  -3.5  1.26  2.7E-03 -7.55E-01  3.06  1.40  7.4E-01    0.700    0.05 -1.0E+02
     7  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
     8  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
     9  0.189  0.028  0.322  -3.5  1.26  5.7E-04 -1.08E+00  3.88  1.30  5.5E-01    0.370    0.05 -1.0E+02
    10  0.189  0.028  0.322  -3.5  1.26  5.7E-05 -1.08E+00  3.88  1.00  5.5E-01    0.037    0.05 -1.0E+02
--
---------------Rockl Properties (Brooks & Corey/van Genuchten combined)----------------------------------------
--     rkcp   etresidpor    beta  alpha ksat     PE         B     n     vg-alpha fracks   imbibe  potir
--                                    Kg s /m^3J/Kg                    1/(J/Kg) Kg s /m^3mm/dy   J/Kg
    129
      1     0.35    0.065   0.366    -1.5    1.26   0.0011    -0.93     3.6    1.25    0.52  0.00056     500    -100
      2    0.043    0.005   0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.41    -100
      3    0.401     0.07   0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
      4    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
      5    0.401     0.07   0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
      6    0.043    0.005   0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.41    -100
      7    0.401     0.07   0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
      8    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
      9    0.401     0.07   0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
     10    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
     11    0.105     0.02    0.11    -1.5    1.26  4.0E-09       -6    3.92    1.47      64  2.4E-07    0.06    -100
     12    0.388    0.039   0.393    -1.5    1.26 0.000053     -242    4.57    1.38    2400 0.000053    2.74    -100
     13     0.43    0.044   0.435    -1.5    1.26   0.0004    -1790    6.81    1.23   18000   0.0004   13.83    -100
     14    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
     15    0.401     0.07   0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
     16    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
     17    0.043    0.005   0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.35    -100
     18    0.248     0.01   0.253    -1.5    1.26  3.8E-06    -82.3    2.71    1.84     830  3.9E-06    3.34    -100
     19    0.159     0.01   0.164    -1.5    1.26  1.2E-06     -140    3.61    1.53    1400  1.3E-06    1.13    -100
     20    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     21    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     22    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     23    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     24    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     25    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
     26    0.077    0.011   0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
```

27	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
28	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
29	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
30	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
31	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
32	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
33	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
34	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
35	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
36	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
37	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
38	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
39	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
40	0.494	0.05	0.499	-1.5	1.26	0.000088	-341	4.19	1.43	3400	0.000088	75.62	-100
41	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
42	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
43	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
44	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-26.9	5.6	1.29	270	2.1E-07	0.09	-100
45	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
99	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
46	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
47	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
48	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
49	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
50	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
51	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
52	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
53	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
54	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
55	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
56	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
57	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
58	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
59	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
60	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
61	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
62	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
63	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
64	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
65	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
66	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
67	0.031	0.018	0.036	-1.5	1.26	5.0E-09	-0.6	3.39	1.58	10	2.4E-07	0.17	-100
68	0.168	0.021	0.173	-1.5	1.26	7.3E-08	-125	5.36	1.31	1300	1.3E-07	0.07	-100
69	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.02	-100
70	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
71	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.01	-100
72	0.317	0.023	0.322	-1.5	1.26	3.8E-08	-181	4.01	1.46	1800	9.9E-08	1.65	-100
73	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
74	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
75	0.281	0.051	0.286	-1.5	1.26	1.7E-08	-17.5	4.01	1.45	180	7.8E-08	0.02	-100
76	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100

77	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
201	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
202	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.06	-100
203	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
204	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	120	1.4E-07	500	-100
205	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
206	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
207	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
208	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
209	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
210	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
211	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
212	0	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
213	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
214	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
301	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
302	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
303	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
304	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
305	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
306	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
307	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
308	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
309	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
310	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
311	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
312	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
313	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
314	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
315	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
316	0.159	0.01	0.164	-1.5	1.26	1.2E-06	-140	3.61	1.53	1400	1.3E-06	1.13	-100
317	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
318	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
319	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
320	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
321	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
322	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
323	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
324	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
325	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
326	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
327	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
328	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
329	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
330	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
331	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
332	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
333	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
334	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
335	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
336	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100

345	0.097	0.01	0.102	-1.5	1.26	1.3E-07	-154.6	5.02	1.56	1560	2.9E-07	0.24	-100

Table III-2. Sample Input Control File for INFIL Version 2.a1 Produced by Manually Substituting Sampled Parameter Values

```
INFIL2a1.ctl: Solitario East, Tule Lake, ww1-2a1-gm1-k2-w20 (11/18/1999)
1   0.00000001                                  IROUT(1=coupled, 0=uncoupled, -1=flow routing off, -2=infil off),
                                                            IFRTOL
2   2     2.16                                      ISNOW, ISNWMOD, SNOPAR1
3   0.157 0.3                                     ISUBLIM, SUBPAR1, SUBPAR2
0   1.                                            IPPTTEST (1 for testing), PPTTEST = constant
0   .0                                            IETTEST (1 for testing), ETTEST = constant
30.0                                                  CELSIZE (node spacing (meters): using for flow volume calculations)
544721.0  4072203.0                               xcfs,ycfs: coords for discharge cell
1950  1    1996    366   1   5                   yr1 = start year, dnn1 = start day, yr2 = end year, dn2 = end day
1.12   0.901  5.48  2.23                          multipliers ( pptfact, etfact, imbfact, sksfact)
1.13  0    0.4                                    SDFACT (soil depth multiplier), IVEGC (set to 1 for map data), FVEGC
                                                            use if IVEGC = 0)
1.0   0.8   0.8    0.2                            ROOTF1,ROOTF2,ROOTF3,ROOTF4
1.0   1.0   0.5    0.1                            MAXWGT1, MAXWGT2, MAXWGT3, MAXWGT4
0.3   2.0   3.75   2.0                            RDEPTH1,RDEPTH2,RDEPTH3,RDEPTH4,RDEPTHF
2.95E-02 1.0  .304                                RKPOR, RKMMFACT, FLAREA
1   1    0                                        INFMOD, ETMOD, RUNMOD
 -10.61  1.102                                  BARSOIL1, BARSOIL2:  bare soil et parameters
3   17.3  11.74                                   IAIRTEMP=1 for new air temp model, ATEMP1 = avg. air temp,
                                                            ATEMP2=air temp seasonal deviation
2                                                 HSTEP: time step for PET model (hours)
5  181  3                                         PPTYUC (=5 diminished elev. correlation, =2 for 4JA , =1 for simple
                                                            elevation transfer), AAPREPX, IPPTDAT
Tulelake.inp                                      input file name:  daily precip
0      1    0.5      1     1                       dpthflag, irtz, delvwcf, moistcr, fracmod
0      1.75                                     IVWCFLG,   vwcfact
ww1.w20                                           input file name:  map parameters (*.inp)
-1
ww1-gm1.2a1                                       dayall: average daily mass balance terms
1    1                                          ndaymap, imap
70   1995
ww1-gm1.2a2
ww1-gm1.2a3                                       output file name:  daily mass balance
ww1-gm1.2a4                                       output file name:  total daily fluxes
1      1                                          dbgflag, dbgflag2
ww1-gm1.2a5                                       output file name:  annual mass balance terms
0                                                 IDEBUG:  debugging option parameter
ww1-gm1.2a6                                       new debug file
ww1-gm1                                           map output:  annual totals or mult-year averages
--
--------------- Parameters for dynamic root-zone function -------------------------------------
--     depth  rtza   rtzb   rtzc  rtzd  bsoil    delvwc
```

```
--      m
     4
     1    0.5     15      3     3     2     0.30      0.50
     2    1.5     15      4     2     2     0.20      0.50
     3    4.5     10    1.5     1     2     0.10      0.50
     4    6.0     10    1.5     1     2     0.05      0.25
--
--------------- Soil Properties (Brooks & Corey/van Genuchten combined) ----------------------------------------
--      fdcp   etresidpor    beta  alpah ksat     PE        B     n     vg-alpha sorp     SOILP   potis
--                                         Kg s /m^3J/Kg                 1/(J/Kg) ???              J/Kg
    10
     1  0.242  0.054  0.366  -3.5  1.26  5.6E-04 -1.19E+00  4.72  1.24  5.2E-01    0.390    0.05 -1.0E+02
     2  0.173  0.023  0.315  -3.5  1.26  1.2E-03 -9.41E-01  3.70  1.31  6.2E-01    0.500    0.05 -1.0E+02
     3  0.163  0.017  0.325  -3.5  1.26  1.3E-03 -8.60E-01  3.36  1.36  6.6E-01    0.510    0.05 -1.0E+02
     4  0.073  0.002  0.281  -3.5  1.26  3.8E-03 -6.22E-01  2.18  1.62  8.7E-01    0.700    0.05 -1.0E+02
     5  0.200  0.035  0.330  -3.5  1.26  6.7E-04 -1.07E+00  4.14  1.78  5.6E-01    0.400    0.05 -1.0E+02
     6  0.150  0.011  0.339  -3.5  1.26  2.7E-03 -7.55E-01  3.06  1.40  7.4E-01    0.700    0.05 -1.0E+02
     7  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
     8  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
     9  0.189  0.028  0.322  -3.5  1.26  5.7E-04 -1.08E+00  3.88  1.30  5.5E-01    0.370    0.05 -1.0E+02
    10  0.189  0.028  0.322  -3.5  1.26  5.7E-05 -1.08E+00  3.88  1.00  5.5E-01    0.037    0.05 -1.0E+02
--
---------------Rockl Properties (Brooks & Corey/van Genuchten combined)----------------------------------------
--      rkcp   etresidpor    beta  alpha ksat     PE        B     n     vg-alpha fracks   imbibe  potir
--                                         Kg s /m^3J/Kg                 1/(J/Kg) Kg s /m^3mm/dy   J/Kg
    129
      1    0.35    0.065    0.366    -1.5    1.26   0.0011   -0.93    3.6    1.25    0.52  0.00056     500    -100
      2   0.043    0.005    0.048    -1.5    1.26  1.6E-07   -88.1   6.43    1.25     890  3.7E-07    0.41    -100
      3   0.401     0.07    0.406    -1.5    1.26 0.000052   -4150   6.78    1.23   42000 0.000052   46.66    -100
      4   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.09    -100
      5   0.401     0.07    0.406    -1.5    1.26 0.000052   -4150   6.78    1.23   42000 0.000052   46.66    -100
      6   0.043    0.005    0.048    -1.5    1.26  1.6E-07   -88.1   6.43    1.25     890  3.7E-07    0.41    -100
      7   0.401     0.07    0.406    -1.5    1.26 0.000052   -4150   6.78    1.23   42000 0.000052   46.66    -100
      8   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.09    -100
      9   0.401     0.07    0.406    -1.5    1.26 0.000052   -4150   6.78    1.23   42000 0.000052   46.66    -100
     10   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.09    -100
     11   0.105     0.02     0.11    -1.5    1.26  4.0E-09      -6   3.92    1.47      64  2.4E-07    0.06    -100
     12   0.388    0.039    0.393    -1.5    1.26 0.000053    -242   4.57    1.38    2400 0.000053    2.74    -100
     13    0.43    0.044    0.435    -1.5    1.26   0.0004   -1790   6.81    1.23   18000   0.0004   13.83    -100
     14   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.09    -100
     15   0.401     0.07    0.406    -1.5    1.26 0.000052   -4150   6.78    1.23   42000 0.000052   46.66    -100
     16   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.09    -100
     17   0.043    0.005    0.048    -1.5    1.26  1.6E-07   -88.1   6.43    1.25     890  3.7E-07    0.35    -100
     18   0.248     0.01    0.253    -1.5    1.26  3.8E-06   -82.3   2.71    1.84     830  3.9E-06    3.34    -100
     19   0.159     0.01    0.164    -1.5    1.26  1.2E-06    -140   3.61    1.53    1400  1.3E-06    1.13    -100
     20   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.06    -100
     21   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.06    -100
     22   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.06    -100
     23   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.06    -100
     24   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.06    -100
     25   0.077    0.011    0.082    -1.5    1.26  5.4E-09     -12   3.04    1.69     120  1.4E-07    0.06    -100
```

26	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
27	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
28	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
29	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
30	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
31	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
32	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
33	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
34	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
35	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
36	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
37	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
38	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
39	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
40	0.494	0.05	0.499	-1.5	1.26	0.000088	-341	4.19	1.43	3400	0.000088	75.62	-100
41	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
42	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
43	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
44	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-26.9	5.6	1.29	270	2.1E-07	0.09	-100
45	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
99	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
46	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
47	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
48	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
49	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
50	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
51	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
52	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
53	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
54	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
55	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
56	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
57	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
58	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
59	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
60	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
61	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
62	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
63	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
64	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
65	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
66	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
67	0.031	0.018	0.036	-1.5	1.26	5.0E-09	-0.6	3.39	1.58	10	2.4E-07	0.17	-100
68	0.168	0.021	0.173	-1.5	1.26	7.3E-08	-125	5.36	1.31	1300	1.3E-07	0.07	-100
69	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.02	-100
70	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
71	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.01	-100
72	0.317	0.023	0.322	-1.5	1.26	3.8E-08	-181	4.01	1.46	1800	9.9E-08	1.65	-100
73	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
74	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
75	0.281	0.051	0.286	-1.5	1.26	1.7E-08	-17.5	4.01	1.45	180	7.8E-08	0.02	-100

76	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
77	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
201	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
202	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.06	-100
203	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
204	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	120	1.4E-07	500	-100
205	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
206	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
207	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
208	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
209	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
210	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
211	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
212	0	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
213	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
214	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
301	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
302	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
303	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
304	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
305	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
306	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
307	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
308	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
309	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
310	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
311	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
312	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
313	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
314	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
315	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
316	0.159	0.01	0.164	-1.5	1.26	1.2E-06	-140	3.61	1.53	1400	1.3E-06	1.13	-100
317	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
318	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
319	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
320	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
321	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
322	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
323	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
324	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
325	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
326	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
327	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
328	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
329	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
330	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
331	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
332	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
333	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
334	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
335	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100

336	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
345	0.097	0.01	0.102	-1.5	1.26	1.3E-07	-154.6	5.02	1.56	1560	2.9E-07	0.24	-100

Table III-3 Listing of sampled parameter values in Realization #001 as shown by GROPECDB. (Element Block 1 FOOTPRNT contains the sampled values).

```
                                   GROPECDB_PA96

 GGGGG  RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB         PPPPPP
GG  GGG RRRRR   OO   OO PP      EE      CC      DD   DD BB   BB         PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB         PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB _______ PP
```

GROPECDB_PA96 Version 2.12
PROD PA96 Built 06/27/96
Sponsored by Amy Gilkey

Run on 01/19/00 at 12:39:40
Run on ALPHA AXP BEATLE OpenVMS V7.1

Database: I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.GLACIAL]POSTLHS_1_R001.CDB;2
Written on: 12/01/99 18:02:24

CAMDAT Version: 1 (EXODUS Version: 1)

PROPERTIES

Element Block 1) "FOOTPRNT" 1=ID 75 elements (1..75)
8-node 5 attributes 13 properties

BRPOROS	BRZDEPTH	SOILDEPM	PRECIPM	POTETMUL	BRPERM
SOILPERM	ETCOEFFA	ETCOEFFB	FLAREA	ATEMP1	SNOPAR1
SUBPAR1					
2.95100E-02	3.74700E+00	1.12800E+00	1.11500E+00	9.01500E-01	5.48200E+00
2.22800E+00	1.06100E+01	1.10200E+00	3.04400E-01	1.73000E+01	2.15800E+00
1.57100E-01					

Element Block 2) "OUTSIDE " 2=ID 25 elements (76..100)

8-node 5 attributes 13 properties

BRPOROS	BRZDEPTH	SOILDEPM	PRECIPM	POTETMUL	BRPERM
SOILPERM	ETCOEFFA	ETCOEFFB	FLAREA	ATEMP1	SNOPAR1
SUBPAR1					
2.00000E-02	1.50000E+00	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
1.00000E+00	-1.50000E+00	1.26000E+00	5.00000E-01	1.73000E+01	1.78000E+00
1.00000E-01					

Table III-4. Summary Output File from Executing INFIL Using PREINFIL to produce an Input User Control File.

```
INFIL2a1.ctl: Solitario East, Tule Lake, ww1-2a1-gm1-k2-w20 (11/18/1999)
Daily precipitation input:              Tulelake.inp
Watershed modeling domain parameters:   ww1.w20
Average daily mass balance terms:       ww1-gm1.2a1
24-hour mass balance map:               ww1-gm1.2a2
Annual mass balance map:                ww1-gm1.1
Average annual mass balance map:        INFILMN_WW1_R001.OUT
Summary statistics output:              INFILSM_WW1_R001.OUT
Debugging output:                       ww1-gm1.2a5


Total number of days read in   =    12053
Total daily precip             =   9660.7
Average annual precip (mm)     =    292.8
Maximum daily precip (mm)      =     51.3


TOTAL NUMBER OF LOCATIONS     =     1325
AVERAGE ELEVATION OF SAMPLE   =   1375.1
MAXIMUM ELEVATION OF SAMPLE   =   1477.0
MINIMUM ELEVATION OF SAMPLE   =   1244.0
AVERAGE SOIL DEPTH (M)        =    0.978
AVERAGE SLOPE OF SAMPLE       =     15.7
MAXIMUM SLOPE OF SAMPLE       =     31.0
NUMBER OF ACTIVE LOCATIONS    =     1121


  Yr  Dy    Precip      Rain      Snow  Sn-cover  Snowmelt   Sublim       PET     PETRS  Evapotrs  Del-soil   Net-inf    Runoff  Out-flow    Mass-
balance
1950 365   327.368   247.180    80.188   518.376    64.458    6.440   740.934   588.043   237.019    74.619     0.000     0.000     0.000  -
0.116272812E-07
1951 365   240.805   133.410   107.395  1050.796    67.072    9.551   748.068   593.705   190.816     8.893     0.773     0.000     0.000  -
0.684339702E-07
1952 366   307.920   259.992    47.928   447.559    78.249    5.716   719.220   570.810   296.593    19.635    22.014     6.051     0.000  -
0.264484373E-06
1953 365   416.968   346.823    70.145   339.105    64.321    7.574   707.794   561.741   315.174    54.204    41.546     6.212     0.000  -
0.508485369E-06
1954 365   240.805   152.043    88.761   721.181    64.230    8.407   720.911   572.151   241.519   -69.759    44.734    12.262     0.000  -
0.849553510E-06
1955 365   288.921   137.320   151.600  1844.158   102.263    9.147   738.829   586.372   192.625    44.914     1.773     0.570     0.000  -
0.438779343E-07
1956 366   351.539   319.022    32.517  2744.499    83.849    7.256   734.147   582.656   309.279    17.799    74.861    48.234     1.203  -
0.445807547E-06
1957 365   432.032   400.401    31.631    92.526    28.441    3.190   749.154   594.567   324.425    46.042    58.374    13.873     0.000  -
0.832975722E-06
1958 365   386.839   338.501    48.338   184.839    35.584    4.291   723.782   574.430   374.525   -56.217    55.777    22.067     0.000  -
0.156226674E-05
```

```
48  1959 365   136.703   121.848   14.855   118.058   19.872   3.445   765.636   607.647   220.895   -82.705   3.530   0.000   0.000  -
49  0.187680335E-05
50  1960 366   319.949   301.681   18.268    46.664   15.739   2.529   750.060   595.286   248.931    63.339   5.150   0.028   0.000  -
51  0.538488827E-07
52  1962 365   383.916   344.603   39.313   352.410   34.481   4.833   736.569   584.579   276.128    54.420  48.535  15.961   0.000  -
53  0.252647791E-06
54  1963 365   264.525   202.033   62.492  1285.523   45.994  10.539   718.305   570.083   259.738   -51.876  40.165   3.419   0.000  -
55  0.679942992E-06
56  1964 366   428.659   252.576  176.084  1379.128   54.457   4.992   731.361   580.445   278.515    12.023  16.495   2.263   0.000  -
57  0.916769591E-06
58  1966 365   263.401   198.532   64.870  3496.565  147.752  10.580   748.145   593.766   247.151    34.955  64.178  30.882   0.000  -
59  0.136124145E-05
60  1967 365   233.835   197.682   36.153   537.506   60.049   4.990   755.938   599.951   292.812   -82.392  47.311   6.070   0.000  -
61  0.287281606E-06
62  1968 366   329.729   311.471   18.259    53.384   13.964   4.034   743.881   590.381   283.441    40.148   1.845   0.881   0.000  -
63  0.455076946E-06
64  1969 365   377.845   264.531  113.314  1294.414   33.754   4.306   749.990   595.230   276.185    -9.894  31.994  12.672   0.000  -
65  0.733306847E-06
66  1970 365   292.631   197.782   94.848  1641.513  147.064   9.340   739.280   586.730   235.241    36.249  73.356  37.382   0.000  -
67  0.127013259E-05
68  1971 365   318.937   272.421   46.516   423.788   55.306   5.302   741.081   588.160   302.568    -7.868  33.027   2.133   0.000  -
69  0.161401473E-05
70  1974 365   189.653   145.123   44.531   232.463   36.380   6.180   742.654   589.408   214.243   -47.742  15.002   0.528   0.000  -
71  0.914091675E-07
72  1975 365   303.198   254.524   48.674   169.348   43.390   6.045   736.015   584.139   277.469     8.366  12.078   0.350   0.000  -
73  0.263568769E-06
74  1976 366   257.668   219.856   37.812   668.688   32.468   6.664   728.463   578.145   270.515   -26.376   8.185   0.618   0.000  -
75  0.408913383E-06
76  1977 365   355.474   261.063   94.411   638.011   38.991   3.691   739.356   586.791   270.753    27.269   2.031   0.106   0.000  -
77  0.544388076E-06
78  1978 365   292.293   234.584   57.709  1134.339  103.323   6.116   752.387   597.133   273.794    -5.961  70.074  38.256   0.000  -
79  0.954334071E-06
80  1979 365   355.361   308.980   46.381   136.882   41.819   2.950   747.073   592.915   251.077    76.559  23.138   7.664   0.000  -
81  0.890559467E-07
82  1980 366   292.293   265.029   27.265   160.762   25.550   2.639   759.633   602.883   296.760   -71.858  65.703  29.325   0.000  -
83  0.730462317E-06
84  1981 365   442.599   299.707  142.892   694.287   95.155   9.846   722.279   573.237   261.592    85.962  47.308  13.348   0.000  -
85  0.105218025E-05
86  1982 365   372.562   332.566   39.995   588.422   66.373   5.585   722.813   573.661   335.504    -8.368  71.585  27.398   0.218  -
87  0.181866569E-05
88  1983 365   511.064   338.886  172.178  1811.492   78.738   9.325   720.650   571.945   301.493    17.793  98.338  38.298   0.000  -
89  0.268142379E-05
90  1987 365   270.821   219.715   51.106  2953.074  121.743   9.380   751.721   596.604   298.932   -47.595  86.171  42.315   3.950  -
91  0.600794386E-06
92  1995 365   467.669   406.737   60.932   396.442   66.445   5.202   729.251   578.770   316.553    56.781  99.068  39.465   0.740  -
93  0.145672736E-05
94  1996 366   412.920   276.575  136.345   860.166   99.629   9.452   730.884   580.066   310.163   -38.762 104.463  49.617   0.381  -
95  0.252140816E-05
96
97
98  Global Summary Statistics (mm/year):
```

```
99   ----------------------------------------
100
101  Precipitation..................      329.306884
102  Rain...........................      259.496179
103  Snowfall.......................       69.810705
104  Snow-cover.....................      879.302133
105  Snow-melt......................       62.634724
106  Sublimation....................        6.349767
107  Potential Evapotranspiration....     737.780869
108  Actual Evapotranspiration.......     275.230822
109  Change in Soil Moisture........        5.230281
110  Net Infiltration...............       41.473078
111  Runoff Generation..............       15.401762
112  Cumulative Daily Run-on........      126.673640
113  Outflow........................        0.196722
114  Average Mass Balance Error......  -0.100983E-11
115  Average Max Daily Error (mm/dy).  -0.753407E-13
```

Table III-5. Summary Output File from Executing INFIL Using Manual Substitutions to produce an Input User Control File.

```
INFIL2a1.ctl: Solitario East, Tule Lake, ww1-2a1-gm1-k2-w20 (11/18/1999)
Daily precipitation input:              Tulelake.inp
Watershed modeling domain parameters:   ww1.w20
Average daily mass balance terms:       ww1-gm1.2a1
24-hour mass balance map:               ww1-gm1.2a2
Annual mass balance map:                ww1-gm1.1
Average annual mass balance map:        ww1-gm1.2a4
Summary statistics output:              ww1-gm1.2a3
Debugging output:                       ww1-gm1.2a5


Total number of days read in   =    12053
Total daily precip             =   9660.7
Average annual precip (mm)     =    292.8
Maximum daily precip (mm)      =     51.3


TOTAL NUMBER OF LOCATIONS    =     1325
AVERAGE ELEVATION OF SAMPLE  =   1375.1
MAXIMUM ELEVATION OF SAMPLE  =   1477.0
MINIMUM ELEVATION OF SAMPLE  =   1244.0
AVERAGE SOIL DEPTH (M)       =    0.978
AVERAGE SLOPE OF SAMPLE      =     15.7
MAXIMUM SLOPE OF SAMPLE      =     31.0
NUMBER OF ACTIVE LOCATIONS   =     1121
```

Yr	Dy	Precip	Rain	Snow	Sn-cover	Snowmelt	Sublim	PET	PETRS	Evapotrs	Del-soil	Net-inf	Runoff	Out-flow	Mass-balance
1950	365	327.368	247.180	80.188	518.376	64.458	6.440	740.934	588.043	237.019	74.619	0.000	0.000	0.000	-0.116272812E-07
1951	365	240.805	133.410	107.395	1050.796	67.072	9.551	748.068	593.705	190.816	8.893	0.773	0.000	0.000	-0.684339702E-07
1952	366	307.920	259.992	47.928	447.559	78.249	5.716	719.220	570.810	296.593	19.635	22.014	6.051	0.000	-0.264484373E-06
1953	365	416.968	346.823	70.145	339.105	64.321	7.574	707.794	561.741	315.174	54.204	41.546	6.212	0.000	-0.508485369E-06
1954	365	240.805	152.043	88.761	721.181	64.230	8.407	720.911	572.151	241.519	-69.759	44.734	12.262	0.000	-0.849553510E-06
1955	365	288.921	137.320	151.600	1844.158	102.263	9.147	738.829	586.372	192.625	44.914	1.773	0.570	0.000	-0.438779343E-07
1956	366	351.539	319.022	32.517	2744.499	83.849	7.256	734.147	582.656	309.279	17.799	74.861	48.234	1.203	-0.445807547E-06
1957	365	432.032	400.401	31.631	92.526	28.441	3.190	749.154	594.567	324.425	46.042	58.374	13.873	0.000	-0.832975722E-06
1958	365	386.839	338.501	48.338	184.839	35.584	4.291	723.782	574.430	374.525	-56.217	55.777	22.067	0.000	-0.156226674E-05
1959	365	136.703	121.848	14.855	118.058	19.872	3.445	765.636	607.647	220.895	-82.705	3.530	0.000	0.000	-0.187680335E-05
1960	366	319.949	301.681	18.268	46.664	15.739	2.529	750.060	595.286	248.931	63.339	5.150	0.028	0.000	-0.538488827E-07
1962	365	383.916	344.603	39.313	352.410	34.481	4.833	736.569	584.579	276.128	54.420	48.535	15.961	0.000	-0.252647791E-06
1963	365	264.525	202.033	62.492	1285.523	45.994	10.539	718.305	570.083	259.738	-51.876	40.165	3.419	0.000	-0.679942992E-06
1964	366	428.659	252.576	176.084	1379.128	54.457	4.992	731.361	580.445	278.515	12.023	16.495	2.263	0.000	-0.916769591E-06
1966	365	263.401	198.532	64.870	3496.565	147.752	10.580	748.145	593.766	247.151	34.955	64.178	30.882	0.000	-0.136124145E-05
1967	365	233.835	197.682	36.153	537.506	60.049	4.990	755.938	599.951	292.812	-82.392	47.311	6.070	0.000	-0.287281606E-06
1968	366	329.729	311.471	18.259	53.384	13.964	4.034	743.881	590.381	283.441	40.148	1.845	0.881	0.000	-0.455076946E-06
1969	365	377.845	264.531	113.314	1294.414	33.754	4.306	749.990	595.230	276.185	-9.894	31.994	12.672	0.000	-0.733306847E-06
1970	365	292.631	197.782	94.848	1641.513	147.064	9.340	739.280	586.730	235.241	36.249	73.356	37.382	0.000	-0.127013259E-05
1971	365	318.937	272.421	46.516	423.788	55.306	5.302	741.081	588.160	302.568	-7.868	33.027	2.133	0.000	-0.161401473E-05
1974	365	189.653	145.123	44.531	232.463	36.380	6.180	742.654	589.408	214.243	-47.742	15.002	0.528	0.000	-0.914091675E-07
1975	365	303.198	254.524	48.674	169.348	43.390	6.045	736.015	584.139	277.469	8.366	12.078	0.350	0.000	-0.263568769E-06
1976	366	257.668	219.856	37.812	668.688	32.468	6.664	728.463	578.145	270.515	-26.376	8.185	0.618	0.000	-0.408913383E-06
1977	365	355.474	261.063	94.411	638.011	38.991	3.691	739.356	586.791	270.753	27.269	2.031	0.106	0.000	-0.544388076E-06
1978	365	292.293	234.584	57.709	1134.339	103.323	6.116	752.387	597.133	273.794	-5.961	70.074	38.256	0.000	-0.954334071E-06

1979	365	355.361	308.980	46.381	136.882	41.819	2.950	747.073	592.915	251.077	76.559	23.138	7.664	0.000	-0.890559467E-07
1980	366	292.293	265.029	27.265	160.762	25.550	2.639	759.633	602.883	296.760	-71.858	65.703	29.325	0.000	-0.730462317E-06
1981	365	442.599	299.707	142.892	694.287	95.155	9.846	722.279	573.237	261.592	85.962	47.308	13.348	0.000	-0.105218025E-05
1982	365	372.562	332.566	39.995	588.422	66.373	5.585	722.813	573.661	335.504	-8.368	71.585	27.398	0.218	-0.181866569E-05
1983	365	511.064	338.886	172.178	1811.492	78.738	9.325	720.650	571.945	301.493	17.793	98.338	38.298	0.000	-0.268142379E-05
1987	365	270.821	219.715	51.106	2953.074	121.743	9.380	751.721	596.604	298.932	-47.595	86.171	42.315	3.950	-0.600794386E-06
1995	365	467.669	406.737	60.932	396.442	66.445	5.202	729.251	578.770	316.553	56.781	99.068	39.465	0.740	-0.145672736E-05
1996	366	412.920	276.575	136.345	860.166	99.629	9.452	730.884	580.066	310.163	-38.762	104.463	49.617	0.381	-0.252140816E-05

Global Summary Statistics (mm/year):

Precipitation.....	329.306884
Rain.....	259.496179
Snowfall.....	69.810705
Snow-cover.....	879.302133
Snow-melt.....	62.634724
Sublimation.....	6.349767
Potential Evapotranspiration....	737.780869
Actual Evapotranspiration.....	275.230822
Change in Soil Moisture.....	5.230281
Net Infiltration.....	**41.473078**
Runoff Generation.....	15.401762
Cumulative Daily Run-on.....	126.673640
Outflow.....	0.196722
Average Mass Balance Error.....	-0.100983E-11
Average Max Daily Error (mm/dy).	-0.753407E-13

ATTACHMENT IV

LISTING OF SOURCE CODE PREINFIL

(TO SATISFY AP-SI.1Q)

This attachment contains a listing of the software routine PREINFIL developed at Sandia National Laboratories which serves as a preprocessor to produce input user control files for the code INFIL (INFIL V.A_2a.1). Verification of the use of this routine is found in Attachment III.

Table IV-1. Source listing file for PREINFIL routine. Writes a new input control file for Program INFIL. McCurley (299-1282) December 1999.

```
PROGRAM PREINFIL
C***********************************************************************
C
C                                PREINFIL
C                                ------
C
C        The PREINFIL translator extracts (LHS) sampled and fixed
C        parameter data and other auxillary information
C        requested by the user from a Computational DataBase file
C        (.CDB), and sets up the INFIL program input control file
C
C                                MAY 1999
C
C
C                                Authors
C                                -------
C
C                           Ron D. McCurley
C                        Applied Mechanics Division
C                NEW MEXICO ENGINEERING RESEARCH INSTITUTE
C                         University of New Mexico
C                    Albuquerque, NM  87131 (505) 272-7227
C
C
C                              Abstract
C                              --------
C
C   Translates input from a Computational Data Base file and a PREINFIL
C   input control file template to a new input control file that is
recognized
C   by the INFIL code used in the YMP performanxce assessment analysis.
C
C                           Primary Reference
C                           -----------------
C
C
C                            Update History
C                            --------------
C
C
C  Version     Date          Modified by        Changes
C
C  ALPHA 1.0   May,1999      Ron D. McCurley    Original Version
C  ALPHA 1.1   Aug, 1999     Ron D. McCurley    modified for writing new
C                                               filenames to INFIL .ctl
C  ALPHA 1.11  Nov 8, 1999   Ron D. McCurley   modified for modifying
lognormally
C                                               distributed etcoeffa sign to
C                                               be negative
C  ALPHA 1.20  Nov 10, 1999  Ron D. McCurley   modified for adding new
C                                              sampled parameters
C
C                            Disclaimer
```

```
C                                  ----------
C
C       This computer program was prepared as an account of work
C  sponsored by an agency of the United States Government.  Neither the
C  United States Government nor any agency thereof, nor any of their
C  employees, nor any of their contractors, subcontractors,or their
C  employees, makes any warranty, express or implied, or assumes any
C  legal liability or responsibility for the accuracy, completeness,
C  or usefulness of any information, apparatus, product, or process
C  disclosed, or represents that its use would not infringe privately
C  owned rights.  Reference herein to any specific commercial product,
C  process, or service by trade name, trademark, manufacturer, or
C  otherwise, does not necessarily constitute or imply its
C  endorsement, recommendation, or favoring by the United States
C  Government, any agency thereof or any of their contractors or
C  subcontractors.  The views and opinions expressed herein do not
C  necessarily state or reflect those of the United States
C  Government, any agency thereof or any of their contractors
C  or subcontractors.
C
C
C                              Program Modules
C                              ---------------
C
C   CBOUND:   Locates the initial and final non-blank characters in a string.
C   DOECH2:   Entry point in subroutine DOECHO.  This section just reads over
C             portion of a file and does nothing.
C   DOECHO:   Echoes portions of the PREINFIL input control to temporary
files.
C   DYNMEM:   Allocates memory dynamically.
C   EXDATE:   Returns the current date in a character string.
C   EXTIME:   Returns the current time in a character string.
C   EXCPUS:   Returns accumulated CPU time, in seconds.
C   EXPARM:   Returns the operating environment parameters.
C   FILECHO2: Echoes a formatted ASCII inuput file, line by line, to a
C             formatted output file.
C   FFRDLRDS:  Substitute for subroutine FREFLD, input is prompted for, read,
C             and echoed, using specified I/O units.
C   GETINP:   Gets literal input line.
C   GETID:    Gets material and property character IDs.
C   INITDA:   Initializes dynamic array lengths and names.
C   MDCOMP:   Advanced memory management routine used to compress storage.
C   MDDEL:    Memory management routine used to delete dynamic array.
C   MDEROR:   Memory management routine used to print dynamic memory error
C             summary.
C   MDFIND:   Advanced memory management routine used to locate dynamic
array.
C   MDGET:    Memory management routine used to reserve memory block.
C   MDGIVE:   Memory management routine used to release unallocated memory.
C   MDINIT:   Memory management routine used to initialize dynamic memory.
C   MDLIST:   Memory management debugging routine used to list dynamic memory
C             storage tables.
C   MDLONG:   Advanced memory management routine used to adjust dynamic array
C             length.
C   MDRSRV:   Memory management routine used to define dynamic array.
C   MDSTAT:   Memory management routine used to obtain dynamic memory
C             statistics.
```

```
C   MEMADJ:   Adjusts amount of dynamic memory required.
C   OUTPUT:   Writes ASCII information to file NOUTFL about the INFIL input
C             translation run.
C   PREINFIL: PREINFIL program main driver.
C   PREPRO:   PREINFIL input preprocessor which performs QA functions and
calls
C             routine for parsing command line arguements consisting of
C             filenames.  Those files are opened when appropriate or prompted
C             for if not supplied on the command line or set to defaults.
C   PROCTL:   Processes PREINFIL input control file, and determines new
C             parameters needed for INFIL control file.
C   QAABORT:  Prints a fatal error message and then aborts job in case of
error
C             detected in input.
C   QABANNER: Prints the program banner to the terminal or an output file.
C             The banner includes the program name in large letters, the
program
C             description, the version, the revision date, the author, the
C             sponsor, and the current run date and time.
C   QADOEDIS: Prints the Sandia DOE contract statement and the DOE disclaimer
C             to the terminal or an output file.
C   QAFETCH:  Returns the program QA information.  Routine QASETUP must be
C             called to set up the CAMCON_LIB common before QAFETCH is
called.
C   QAPAGE:   Starts a new page (except for the first call) and puts the QA
C             information (program name, run time, etc.) at the top of each
C             page.  At the end of the file, it prints the ending QA
C             information.
C   QAMAXERR: Checksan integer value against a maximum value.  If the value
is
C             over the maximum, an error message is printed and QAABORT is
C             called (if requested).
C   QASETUP:  Called at the start of the program.  It performs initialization
C             details common to all programs.  Specifically, it:
C               Initializes the CPU time,
C               Sets the common area CAMCON_LIB common with the passed
C               information,
C               Sets the current run date and time,
C               Sets the execution machine and operating system,
C               Adds the machine to the program version,
C               Starts a new page on the terminal.
C   RDCMDL:   Reads and parses the command line fields.
C   RDPAR:    Reads the PREINFIL input control file parameter information
C             and returns the variables found.
C   RDINFTMP: Reads the PREINFIL input INFIL template file, stores all
C             record data
C   STRPACK:  Left justifies a string and removes all embedded blanks.
C   TYPRQS:   Outputs a request for a character string and then inputs the
C             character string in an interactive session.
C   WELCOM:   Produces Instructions for main program usage, writes banner and
C             program credits.
C   WRIINF:   Uses Computational Database material properties to write a
C             the revised INFIL input file.
C
C
C
C                         Assumptions and Limitations
```

```
C                          ---------------------------
C
C    Language used is ANSI X3.9-1978 FORTRAN 77 except that comments
C    and Hollerith strings use lowercase characters. INTEGER and REAL
C    variable names are explicitly typed.  Machine dependent coding
C    exists in subroutine WRIINF
C
C                                  Statistics
C                                  ----------
C
C       ALPHA  Version 2.02:
C       12659 lines total,   6730 FORTRAN lines,   5508 FORTRAN statements
C        5930 comment lines, 3487 text lines,
C       total/FORTRAN lines ratio: 1.881
C
C                              Types of data sets
C                              ------------------
C
C  INPUT/
C    ASCII:  in units INASCI, ISECDB
C
C  OUTPUT/
C    ASCII:  in units IOUT, INPINF, ITEMP2, ITEMP3, NOUTFL
C
C                                 Files used
C                                 ----------
C
C  filename           unit     description
C (PREINFILTEMP2.SCR)  2        PREINFIL temporary file #2(STATUS='SCRATCH')
C (PREINFILTEMP3.SCR)  3        "                        #3
C (POSTLHS.CDB)        4        COMPUTATIONAL DATA BASE file
C  SYS$COMMAND         5        Terminal screen/keyboard
C (PREINFIL_SXX.DBG)   6 or 7  (OPTIONAL) PREINFIL diagnostics/debug file
C (PREINFIL_SXX.INP)   8        PREINFIL input control file
C (PREINFIL_SXX.TRN)   9        INFIL input file generated by PREINFIL
C
C*********************************************************************
C234567
      IMPLICIT NONE
      INCLUDE 'IOCOM.INC/LIST'
      INCLUDE 'ARRAY.INC/LIST'
      INCLUDE 'DYNAME.INC/LIST'
      INCLUDE 'FLWCHR.INC/LIST'
      INCLUDE 'INDEX.INC/LIST'
      INCLUDE 'LENGTH.INC/LIST'
      INCLUDE 'LOCAT.INC/LIST'
      INCLUDE 'PARACM.INC/LIST'
      INCLUDE 'CAMCON_COMMON.INC/LIST'

      COMMON /IQACOMMON/ INPROG, INVERS, INDATE, INAUTH, INSPON

      INTEGER  IERR, MORMEMC, MORMEMR
      INTEGER  KI, KF, NUMREC, NUMVAR
      INTEGER  INFDAT(MAXREC,20), NUMFLD(MAXREC)

      REAL     INFIL_VAL(MAXVAR), RDUM, RINFDAT(MAXREC,20)

```

```
225          CHARACTER*12 INPROG
226          CHARACTER*8  INVERS, INDATE
227          CHARACTER*80 INAUTH, INSPON
228          CHARACTER*80 CINFDAT(MAXREC,20)
229          CHARACTER*8  NAMVAR(MAXVAR), INFIL_NAME(MAXVAR)
230          CHARACTER*80 FINF, FOUT, FCDB, FTMP, FUSR
231
232          LOGICAL  WRTOUT
233
234          EXTERNAL BLOCK
235
236    C<><><><><><><><><><><><>
237    C...Begin Procedures...
238    C<><><><><><><><><><><><>
239
240    C     ...Perform routine initializations
241          CALL QASETUP ( INPROG, INVERS, INDATE, INAUTH, INSPON )
242    C
243    C     ...Call for program QA information
244          CALL QAFETCH (INPROG, INVERS, INDATE, RUNDAT, RUNTIM,
245         &   INAUTH, INSPON)
246    C
247    C     ...Prompt USER for program execution control options
248          CALL PREPRO( FUSR, FCDB, FTMP, FINF, FOUT )
249
250    C     ...Show USER the program status
251    C      PRINT *,' *** Begin processing input data ***'
252    C
253    C**** Process the PREINFIL input control file ****
254    C
255    C     ...Read PREINFIL input control file first time
256          OPEN(INASCI,FILE=FUSR,STATUS='UNKNOWN',
257         +     FORM='FORMATTED',READONLY)
258          CALL PROCTL
259
260          CLOSE(INASCI)
261    C     ...read INFIL template and store
262          OPEN(INPTMP,FILE=FTMP,STATUS='UNKNOWN',
263         +     FORM='FORMATTED',READONLY)
264          CALL RDINFTMP(NUMREC, NUMFLD, INFDAT, RINFDAT, CINFDAT)
265    C     ...Show USER the program status
266          PRINT*,' *** Completed processing input data ***'
267          PRINT*,' *** Begin translation to INFIL input file ***'
268    C
269    C     ...Initialize DYNAMIC base arrays
270          CALL MDINIT( Q )
271          CALL MCINIT( CQ )
272
273    C     ...Set-up the DYNAMIC base arrays
274          IERR = 0
275          CALL DBSETUP( Q, CQ, IERR )
276          IF(IERR.NE.0)CALL QAABORT('(PREINFIL) - Setting up DYNAMIC base'//
277         $                          ' arrays %%%')
278
279    C     ...Set debugging/diagnostics file unit
280          CALL DBERRUNI( NOUTFL )
281
```

```
282   C      ...Process input CAMDAT
283          CALL DBIOPEN( ICDB, FCDB, Q, CQ, IERR )
284   C*       OPEN(ICDB,FILE=FCDB,STATUS='UNKNOWN',FORM='UNFORMATTED')
285          CALL PROCDB( MAXC, MAXQ, CQ, IQ, LQ, Q )
286
287   C****    Match COMPUTATIONAL Data Base names and variables with INFIL data
288   and
289   C        translate selected CDB analysis variable values to INFIL values ****
290          CALL TRNDAT(IQ(LNUMPRP), Q(LXMATPR), CQ(LNAMELB), CQ(LNMATPR),
291         +            INFIL_VAL)
292          OPEN(INPINF,FILE=FINF,STATUS='NEW',FORM='FORMATTED')
293   C
294          REWIND INPTMP
295   C      ...Write the new INFIL input file
296          CALL WRIINF(FCDB,NUMREC,NUMFLD,INFDAT,RINFDAT,CINFDAT,INFIL_VAL)
297          CLOSE(ICDB)
298          CLOSE(INPINF)
299          CLOSE(INPTMP)
300   C      ...Show USER the program status
301   C       PRINT*,' *** Completed translation to INFIL input file ***'
302
303   C      ...Successful TRANSLATION, record information
304          CALL OUTPUT
305   C
306   C**** Begin program termination procedures ****
307   C
308          CALL EXCPUS( RDUM )
309          WRITE(NOUTFL,'(/A,F10.4,A)')' CPU time was',RDUM,' seconds'
310          CALL QAPAGE(NOUTFL,'END')
311          CLOSE(NOUTFL)
312          STOP 'PREINFIL Normal Completion'
313   C-----------------------------------------------------------------------
314   C**** END OF PROGRAM PREINFIL {A1MAIN} ****
315   C-----------------------------------------------------------------------
316          END
```

```
*DECK ADDMEM
      SUBROUTINE ADDMEM( MORMEMR, MORMEMC )
C***********************************************************************
C***                                                                 ****
C***        computes  A_D_D_itional  M_E_M_ory  needed  module       ****
C***                  - - -          - - -                           ****
C***********************************************************************
C
C  PURPOSE:       Sets dynamic array lengths, and computes the
C                 amount of memory needed for the REAL base
C                 Q array and CHARACTER*1 base C array.
C
C  AUTHOR:        Jonathan Scott Rath, Esq. [NMERI]
C
C  UPDATED:       17-AUG-1995  --JSR: Deleted TDEC array
C                 26-AUG-1994  --JSR: Adapted for new GENERIC.RDB
C                 14-JUN-1994  --JSR: First Ed.
C
C  CALLED BY:     PREINFIL {main program}
C
C  ENTRY POINTS: ADDMEM1
C
C  ARGUMENTS:
C  ENTRY/
C    --common blocks
C     /IOCOM/ ($include 'IOCOM.INC')
C      NOUTFL = File unit of diagnostics/debug output file
C
C     /DYNAME/ ($include 'DYNAME.INC')
C      CNAMES = Names of the CHARACTER*1 type dynamic arrays
C      DTYPES = Data types of the CNAMES arrays
C      RNAMES = Names of the REAL type dynamic arrays
C
C     /GEOPAR/ ($include 'GEOPAR.INC')
C      NELBLK = Number of ELement BLocKs in mesh (materials)
C      NQAREC = Number of QA records
C      NUQPRP = No. of UniQUe PRoperty names
C
C     /MISCDAT/ (from FORTRAN statement: INCLUDE 'MISCDAT.INC')
C
C  LOCAL/
C      IERR   = INTEGER error flag
C
C  EXIT/
C    --common block
C     /LENGTH/ ($include 'LENGTH.INC')
C      CLENGT = CHARACTER*1 type dynamic array lengths array
C      RLENGT = REAL type dynamic array LENGTHS array
C
C     /INDEX/ ($include 'INDEX.INC')
C      LASCIDX = Last CHARACTER dynamic array index
C      BCDAIDX = Beginning index of CHARACTER dyanamic arrays
C      ECDAIDX = Ending "                                 "
C      LASRIDX = Last REAL dynamic array index
C      BRDAIDX = Beginning index of REAL dyanamic arrays
C      ERDAIDX = Ending "                                 "
```

```
374   C
375   C     --through subroutine call
376   C       MORMEMR= No. of REAL words of memory needed
377   C       MORMEMC= No. of CHARACTER*1 words of memory needed
378   C
379   C*********************************************************************
380   C234567
381         IMPLICIT NONE
382         INCLUDE 'IOCOM.INC'
383         INCLUDE 'DYNAME.INC'
384         INCLUDE 'GEOPAR.INC'
385         INCLUDE 'LENGTH.INC'
386         INCLUDE 'INDEX.INC'
387         INCLUDE 'MISCDAT.INC'
388         INTEGER I, IERR, MORMEMC, MORMEMR
389   C
390   C<><><><><><><><><><><><>
391   C...Begin Procedures...
392   C<><><><><><><><><><><><>
393   C
394   C*********************************************************************
395   C
396   C     PSEUDO ENTRY ADDMEM( MORMEMR, MORMEMC )
397   C
398   C*********************************************************************
399   C
400   C     THIS SECTION COMPUTES MEMORY REQUIRED FOR THE ARRAYS ASSOCIATED
401   C     WITH THE INPUT CAMDAT.
402   C
403   C     (CALLED BY PROCDB)
404   C
405         IERR       = 0
406         MORMEMC    = 0
407         MORMEMR    = 0
408   C     ...Set beginning and ending dynamic array indices
409         BRDAIDX    = 1
410         ERDAIDX    = 4
411         BCDAIDX    = 1
412         ECDAIDX    = 3
413   C     ...Set dynamic REAL array lengths
414   C     ...(1:4) IDEBLK, NUMPRP, IASPRP, XMATPR
415         RLENGT(1)  = NELBLK
416         RLENGT(2)  = RLENGT(1)
417         RLENGT(3)  = NELBLK*NUQPRP
418         RLENGT(4)  = RLENGT(3)
419   C     ...Set dynamic CHARACTER*1 array lengths
420   C     ...(1:3) QAINFO, NAMELB, NMATPR
421         CLENGT(1) = (NQAREC+2)*32
422         CLENGT(2) = NELBLK*8
423         CLENGT(3) = NUQPRP*8
424   C     ...Compute MORMEMR and check for ZERO length (must be at least=1)
425         DO 10 I = BRDAIDX, ERDAIDX, 1
426            IF(RLENGT(I).EQ.0)THEN
427               RLENGT(I) = 1
428            ELSEIF(RLENGT(I).LT.0)THEN
429               IERR = IERR + 1
430               WRITE(NOUTFL,'(A,I3,A)')' ***RLENGT(',I,') < 0***'
```

```
431                WRITE(NOUTFL,'(A,I3,2A)')' RNAMES(',I,')=',RNAMES(I)
432             ENDIF
433             MORMEMR = MORMEMR + RLENGT(I)
434     10   CONTINUE
435          IF(DEBUG)WRITE(NOUTFL,'(A)')' $$$$$$$$$$$$$$$$$$$$$$$$$$$$'//
436         $'$$ ADDMEM $$$$$$$$$$$$$$$$$$$$$$$$$$$$'
437          IF(IERR.GT.0)THEN
438             CALL QAABORT(
439         $   '(ADDMEM) - REAL array size errors detected ***')
440          ELSEIF(MORMEMR.LT.0)THEN
441             WRITE(NOUTFL,*)' **** MORMEMR < 0 in ADDMEM ****'
442             DO 15 I=BRDAIDX,ERDAIDX
443                WRITE(NOUTFL,'(1X,2A,I2,A,I9)')
444         $      RNAMES(I),' RLENGT(',I,')=',RLENGT(I)
445     15      CONTINUE
446             CALL QAABORT(
447         $   '(ADDMEM) - REAL array memory errors detected ***')
448          ENDIF
449   C
450   C      ...Compute MORMEMC and check for ZERO length (must be at least=1)
451          DO 20 I = BCDAIDX, ECDAIDX, 1
452             IF(CLENGT(I).EQ.0)THEN
453                CLENGT(I) = 1
454             ELSEIF(CLENGT(I).LT.0)THEN
455                IERR = IERR + 1
456                WRITE(NOUTFL,'(A,I3,A)')' ***CLENGT(',I,') < 0***'
457                WRITE(NOUTFL,'(A,I3,2A)')' CNAMES(',I,')=',CNAMES(I)
458             ENDIF
459             MORMEMC = MORMEMC + CLENGT(I)
460     20   CONTINUE
461          IF(IERR.GT.0)THEN
462             CALL QAABORT(
463         $   '(ADDMEM) - CHARACTER array size errors detected ***')
464          ELSEIF(MORMEMC.LT.0)THEN
465             WRITE(NOUTFL,*)' **** MORMEMC < 0 in ADDMEM ****'
466             DO 25 I=BCDAIDX,ECDAIDX
467                WRITE(NOUTFL,'(1X,2A,I2,A,I9)')
468         $      CNAMES(I),' CLENGT(',I,')=',CLENGT(I)
469     25      CONTINUE
470             CALL QAABORT(
471         $   '(ADDMEM) - CHARACTER array memory errors detected ***')
472          ENDIF
473          RETURN
474   C
475   C*********************************************************************
476   C
477   C*       ENTRY ADDMEM1( MORMEMR, MORMEMC )
478   C
479   C*********************************************************************
480   C
481   C      THIS SECTION COMPUTES THE MEMORY REQUIRED FOR
482   C      POSTINFIL including net infitration, AND OTHER RELATED DYNAMIC ARRAYS:
483   C
484   C      ...Set beginning and ending dynamic array indices
485   C      ...Set dynamic REAL array lengths
486   C      ...()  MAPGEN
487   C      ...Set dynamic CHARACTER*1 array lengths
```

```
C     ...Check for ZERO length (must be at least=1)

C*      RETURN
C-----------------------------------------------------------------------
C**** END OF SUBROUTINE ADDMEM ****
C-----------------------------------------------------------------------
      END
```

```
495
496   *COMDECK BLOCK
497         BLOCK DATA BLOCK
498   C*********************************************************************
499         INCLUDE 'CAMCON_COMMON.INC/LIST'
500         INCLUDE 'INDEX.INC/LIST'
501         INCLUDE 'DYNAME.INC/LIST'
502
503         INTEGER
504        +ITEMP1,ICDB,ISCREN,NOUTFL,INASCI,INPTMP,INPINF
505
506         CHARACTER*80 LINE1,LINE2,LINE3
507         CHARACTER*12 INPROG
508         CHARACTER*8 INVERS, INDATE
509         CHARACTER*80 INAUTH, INSPON
510
511         COMMON /IOCOM/
512        +ITEMP1,ICDB,ISCREN,NOUTFL,INASCI,INPTMP,INPINF
513
514         COMMON /L1/  LINE1
515         COMMON /L2/  LINE2
516         COMMON /L3/  LINE3
517
518         COMMON /IQACOMMON/ INPROG, INVERS, INDATE, INAUTH, INSPON
519
520   C     ...Dynamic array names (COMMON /DYNAME/)
521         DATA DTYPES/
522        $'INTEGER','INTEGER','LOGICAL','REAL'/
523         DATA RNAMES/
524        $'IDEBLK', 'NUMPRP', 'IASPRP', 'XMATPR'/
525         DATA CNAMES/
526        $'QAINFO','NAMELB','NMATPR'/
527
528         DATA ITEMP1/1/ICDB/4/ISCREN/5/
529        +     INASCI/2/INPTMP/3/INPINF/8/
530
531   C        LASRIDX= Last index of REAL dynamic arrays [INTEGER]
532         DATA LASCIDX/3/LASRIDX/4/
533
534         DATA
535        +LINE1/'The PREINFIL translator extracts new parameter data'/
536        +LINE2/'requested by the user from the computational database file,
537        +'/LINE3/'XXX.CDB, & sets up the INFIL program input control file'/
538
539         DATA INPROG / 'PREINF' /
540         DATA INVERS / 'A1.20' /
541         DATA INDATE / '11/10/99' /
542         DATA INAUTH / 'Ron McCurley' /
543         DATA INSPON / 'Ron McCurley' /
544   *********************************************************************
545         END
```

```
546
547   *DECK,DOECHO
548         SUBROUTINE DOECHO( INFIL, OUTFIL, INREC, BEGIN_ECHO )
549   C***********************************************************************
550   C
551   C   PURPOSE :     Echoes portions of the INFIL input template file
552   C                 to selected file and to debug file
553   C
554   C   AUTHOR:       Ron McCurley
555   C
556   C   UPDATED:      June 8, 1999 --RDM Modified for PREINFIL
557   C
558   C   CALLED BY:    WRIINF
559   C
560   C   ENTRY POINT: DOECH2
561   C
562   C   CALLS:        CBOUND, QAABORT
563   C
564   C   ARGUMENTS:
565   C   ENTRY/
566   C     --through subroutine call
567   C       BEGIN_ECHO = Integer variable describing a record of INASCI
568   C                    where echoing begins
569   C       INREC = Last record read in INASCI
570   C
571   C     --common blocks
572   C      /IOCOM/ ($include 'IOCOM.INC')
573   C       INASCI = File unit of PREINFIL input control file
574   C       ISCREN = "            terminal screen
575   C       NOUTFL = "            diagnostics/debug file
576   C
577   C   LOCAL/
578   C       IL,IR  = Beg. & Ending positions of non-blank characters
579   C       IFILE  = Current file information being echoed to
580   C       IOSTAT = ANSI FORTRAN I/O status integer
581   C
582   C   EXIT/
583   C       none
584   C
585   C***********************************************************************
586   C234567
587         IMPLICIT NONE
588         INCLUDE 'IOCOM.INC/LIST'
589         INTEGER BEGIN_ECHO, I, IERR, IFILE, INFIL, IOSTAT, INREC, IREC,
590        +        OUTFIL
591         CHARACTER*(132) RECORD
592
593   C<><><><><><><><><><><><>
594   C...BEGIN PROCEDURES...
595   C<><><><><><><><><><><><>
596
597   C     ...Set initial values
598         IERR   = 0
599         IREC = INREC + 1
600
601   C     ...Read INFIL input template file
602    100  READ (INFIL,'(A)',IOSTAT=IOSTAT) RECORD
```

```
603          IF (NOUTFL .GT. 0) WRITE (NOUTFL,'(1X,A)') RECORD
604
605          IF(IOSTAT.LT.0)THEN
606   C         ...E-O-F read
607             GOTO 111
608
609          ELSEIF(IOSTAT.GT.0)THEN
610   C         ...Error in while reading
611             IERR = IERR + 1
612
613   C         ...Read next record
614             IREC = IREC + 1
615             GOTO 100
616
617          ELSEIF(IOSTAT.EQ.0)THEN
618
619   C            ...Echo this record to the SPECIFIED file
620             IF ( (IREC .GE. BEGIN_ECHO) .AND. (OUTFIL .NE. NOUTFL) )
621         +     WRITE (OUTFIL,'(A)') RECORD
622
623   C            ...Read next record
624             IREC = IREC + 1
625                GOTO 100
626
627          ENDIF
628
629    111   IF(IERR.GT.0)THEN
630             WRITE(NOUTFL,*)' *** ',IERR,' Errors found in DOECHO ***'
631             CALL QAABORT('DOECHO')
632          ENDIF
633
634          RETURN
635
636   C*********************************************************************
637   C
638   C*      ENTRY DOECH2( RECORD )
639   C
640   C*********************************************************************
641   C
642   C     THIS SECTION JUST READS OVER PORTION OF A FILE AND DOES
643   C     NOTHING (USED BECAUSE THE FREE-FIELD READER CANNOT PROCESS
644   C     THIS INFORMATION DUE TO ITS NON-LIST DIRECTED READ FORMAT !!!)
645
646   C*      IERR = 0
647
648   C* 900  READ (INASCI,'(A)',IOSTAT=IOSTAT) RECORD
649   C*      IF (NOUTFL .GT. 0) WRITE (NOUTFL,'(1X,A)') RECORD
650
651   C*      IF(IOSTAT.LT.0)THEN
652   C        ...E-O-F read
653   C*         GOTO 111
654
655   C*      ELSEIF(IOSTAT.GT.0)THEN
656   C        ...Error in while reading
657   C*         IERR = IERR + 1
658   C        ...Read next record
659   C*         GOTO 900
```

```
660
661   C*       ELSEIF(IOSTAT.EQ.0)THEN
662
663   C*       RETURN
664   C-----------------------------------------------------------------------
665   C**** END OF SUBROUTINE DOECHO ****
666   C-----------------------------------------------------------------------
667         END
```

```
*DECK DYNMEM
      SUBROUTINE DYNMEM( MORMEMR, MEMORY, Q )
C***********************************************************************
C***                                                                 ***
C***          applies D_Y_N_amic  M_E_M_eory  allocation             ***
C***                   - - -       - - -                             ***
C***********************************************************************
C
C   PURPOSE:     Allocates MEMORY DYNAMICally
C
C   AUTHOR:      Jonathan S. Rath
C
C   UPDATED:     19-DEC-1991  --Adapted for REAL dynamic base array
C                01-FEB-1990  --Fixed record lengths <= 72 chars.
C                14-DEC-1989  --Increased INTEGER formats
C                13-JUN-1989  --First Ed.
C
C   CALLED BY:   PROCDB
C                PROCIN
C                PROCQA
C
C   ENTRY POINT: DYNMEM2, DYNMEM3
C
C   CALLS:       MEMADJ
C                MEMADJ2
C                MEMADJ3
C
C   ARGUMENTS:
C   ENTRY/
C     --common block
C      /IOCOM/ ($include 'IOCOM.INC')
C       NOUTFL = File unit of diagnostics/debug output file
C
C     --through subroutine call
C       MORMEMR = Add'l words needed for REAL core allocation (Q)
C       Q       = Dynamic memory REAL base array
C
C   LOCAL/
C       MEMADD = Actual number of words added to core (REAL-Q)
C       MEMOLD = Previous total no. of words allocated to core (Q)
C
C   EXIT/
C     --through subroutine call
C       MEMORY = Total number of words allocated to core (Q-REAL)
C       Q      = Dynamic memory REAL base array
C
C***********************************************************************
C234567
      IMPLICIT NONE
      INCLUDE 'IOCOM.INC'
      INTEGER MEMADD, MEMOLD, MEMORY, MORMEMR
      REAL Q(*)
C
C***********************************************************************
C
C     PSEUDO ENTRY DYNMEM( MORMEMR, MEMORY, Q )
```

```
725   C
726   C*********************************************************************
727   C
728         MEMOLD = MEMORY
729         CALL MEMADJ( MORMEMR, MEMORY, Q )
730         MEMADD = MEMORY - MEMOLD
731         WRITE(NOUTFL,900) MEMADD, 'DYNMEM', MEMORY
732         RETURN
733   C
734   C*********************************************************************
735   C
736         ENTRY DYNMEM2( MORMEMR, MEMORY, Q )
737   C
738   C*********************************************************************
739   C
740   C     THIS SECTION ADJUSTS MEMORY ALLOCATION BY LENGTHENING
741   C     (OR SHORTENING) CERTAIN DYNAMIC ARRAYS
742   C
743         MEMOLD = MEMORY
744         CALL MEMADJ2( MORMEMR, MEMORY, Q )
745         MEMADD = MEMORY - MEMOLD
746         WRITE(NOUTFL,900) MEMADD, 'DYNMEM2', MEMORY
747         IF(MORMEMR.GT.MEMADD)WRITE(NOUTFL,910)MORMEMR,MEMADD,MEMOLD,MEMORY
748         RETURN
749   C
750   C*********************************************************************
751   C
752         ENTRY DYNMEM3( MEMORY, Q )
753   C
754   C*********************************************************************
755   C
756   C     THIS SECTION IS USED TO CONSOLIDATE MEMORY AND/OR FOR
757   C     ADJUSTING MEMORY ALLOCATION DUE TO "PSEUDO-DELETING"
758   C     DYNAMIC ARRAYS.  "PSEUDO-DELETING" MEANS SETTING THE UNUSED
759   C     DYNAMIC ARRAYS TO LENGTH 1.
760   C
761         MEMOLD = MEMORY
762         CALL MEMADJ3( MEMORY, Q )
763         MEMADD = MEMORY - MEMOLD
764         WRITE(NOUTFL,900) MEMADD, 'DYNMEM3', MEMORY
765         RETURN
766   C
767   C---------------  F O R M A T   S T A T E M E N T S  -----------------
768   C
769    900  FORMAT(
770        +//5X,'-----------------------------------------------------',
771        + /5X,'        ',I10,' more words allocated by ',A,
772        + /5X,'        Total allocated is ',I10,
773        + /5X,'-----------------------------------------------------',//)
774    910  FORMAT( //,'***** Possible Error in  D y n a m i c  Memory',
775        +           ' Allocation (Q) *****',
776        +  /,5X,' Min. Additional words REQ"D     = ',I10,' (MORMEMR)',
777        +  /,5X,' Additional words added          = ',I10,' (MEMADD)',
778        +  /,5X,' Previous total word allocation  = ',I10,' (MEMOLD)',
779        +  /,5X,' Current total word allocation   = ',I10,' (MEMORY)',
780        +       //,'*********************************************',
781        +          '******************' )
```

```
782   C
783   C-----------------------------------------------------------------------
784   C**** END OF SUBOUTINE DYNMEM ****
785   C-----------------------------------------------------------------------
786   C
787         END
```

```
788
789   *DECK DYNMEMC
790         SUBROUTINE DYNMEMC( MORMEMC, MEMORY, C )
791   C*********************************************************************
792   C***                                                               ****
793   C***  applies D_Y_N_amic  M_E_M_ory  (C_haracter type) allocation   ****
794   C***          - - -       - - -       -                            ****
795   C*********************************************************************
796   C
797   C   PURPOSE:     Allocates MEMORY DYNAMICally to CHARACTER base array
798   C
799   C   AUTHOR:      Jonathan S. Rath
800   C
801   C   UPDATED:     02-DEC-1991  --First Ed.
802   C
803   C   CALLED BY:   PROCDB
804   C                PROCIN
805   C                PROCQA
806   C
807   C   ENTRY POINT: DYNMEMC2, DYNMEMC3
808   C
809   C   CALLS:       MEMADC
810   C                MEMADJC2
811   C                MEMADJC3
812   C   ARGUMENTS:
813   C   ENTRY/
814   C     --common block
815   C      /IOCOM/ ($include 'IOCOM.INC')
816   C       NOUTFL = File unit of diagnostics/debug output file
817   C
818   C     --through subroutine call
819   C       MORMEMC = Add'l words needed for CHAR core allocation (C)
820   C       C       = Dynamic memory CHARACTER*1 base array
821   C
822   C   LOCAL/
823   C       MEMADD = Actual number of words added to core (CHAR-C)
824   C       MEMOLD = Previous total no. of words allocated to core (C)
825   C
826   C   EXIT/
827   C     --through subroutine call
828   C       MEMORY = Total number of words allocated to core (C-CHAR)
829   C       C      = Dynamic memory CHARACTER*1 base array
830   C
831   C*********************************************************************
832   C234567
833         IMPLICIT NONE
834         INCLUDE 'IOCOM.INC'
835         INTEGER MEMADD, MEMOLD, MEMORY, MORMEMC
836         CHARACTER*(1) C(*)
837   C
838   C*********************************************************************
839   C
840   C     PSEUDO ENTRY DYNMEMC( MORMEMC, MEMORY, C )
841   C
842   C*********************************************************************
843   C
844         MEMOLD = MEMORY
```

```
845          CALL MEMADJC( MORMEMC, MEMORY, C )
846          MEMADD = MEMORY - MEMOLD
847          WRITE(NOUTFL,900) MEMADD, 'DYNMEMC', MEMORY
848          RETURN
849    C
850    C*********************************************************************
851    C
852          ENTRY DYNMEMC2( MORMEMC, MEMORY, C )
853    C
854    C*********************************************************************
855    C
856    C     THIS SECTION ADJUSTS MEMORY ALLOCATION BY LENGTHENING
857    C     (OR SHORTENING) CERTAIN DYNAMIC ARRAYS
858    C
859          MEMOLD = MEMORY
860          CALL MEMADJC2( MORMEMC, MEMORY, C )
861          MEMADD = MEMORY - MEMOLD
862          WRITE(NOUTFL,900) MEMADD, 'DYNMEMC2', MEMORY
863          IF(MORMEMC.GT.MEMADD)WRITE(NOUTFL,910)MORMEMC,MEMADD,MEMOLD,MEMORY
864          RETURN
865    C
866    C*********************************************************************
867    C
868          ENTRY DYNMEMC3( MEMORY, C )
869    C
870    C*********************************************************************
871    C
872    C     THIS SECTION CAN BE USED FOR MEMORY CONSOLIDATION AND/OR FOR
873    C     ADJUSTING MEMORY ALLOCATION DUE TO "PSEUDO-DELETING"
874    C     DYNAMIC ARRAYS.  "PSEUDO-DELETING" MEANS SETTING THE UNUSED
875    C     DYNAMIC ARRAYS TO LENGTH 1.
876    C
877          MEMOLD = MEMORY
878          CALL MEMADJC3( MEMORY, C )
879          MEMADD = MEMORY - MEMOLD
880          WRITE(NOUTFL,900) MEMADD, 'DYNMEMC3', MEMORY
881          RETURN
882    C
883    C---------------  F O R M A T   S T A T E M E N T S  -----------------
884    C
885     900  FORMAT(
886         +//5X,'-------------------------------------------------------',
887         + /5X,' ',I10,' more CHAR*1 words allocated by ',A,
888         + /5X,'            Total allocated is ',I10,
889         + /5X,'-------------------------------------------------------',//)
890     910  FORMAT( //,'***** Possible Error in  D y n a m i c  Memory',
891         +            ' Allocation (C) *****',
892         +  /,5X,' Min. Additional words REQ"D     = ',I10,' (MORMEMC)',
893         +  /,5X,' Additional words added          = ',I10,' (MEMADD)',
894         +  /,5X,' Previous total word allocation  = ',I10,' (MEMOLD)',
895         +  /,5X,' Current total word allocation   = ',I10,' (MEMORY)',
896         +       //,'*********************************************',
897         +          '******************' )
898    C
899    C-----------------------------------------------------------------------
900    C**** END OF SUBOUTINE DYNMEMC ****
901    C-----------------------------------------------------------------------
```

```
902     C
903             END
```

```
904
905   *DECK,ECHOREC
906         SUBROUTINE ECHOREC( INFILUNT, OUTFILUNT, INREC )
907   C*********************************************************************
908   C
909   C   PURPOSE :     Echoes single record of the FILE corresponding to input
910   C                 INFILUNT to the file corresponding to output OUTFILUNT
911   C
912   C   AUTHOR:       Ron McCurley
913   C
914   C   UPDATED:      25 May, 1999 --initial version for PREINFIL
915   C
916   C   CALLED BY:    WRIINF
917   C
918   C   CALLS:        QAABORT
919   C
920   C   ARGUMENTS:
921   C   ENTRY/
922   C     --through subroutine call
923   C       LASTREC = integer variable describing a location of last record
924   C                 read in INFILUNT
925   C       INREC = integer variable pointing to location of record in INFILUNT
926   C
927   C     --common blocks
928   C      /IOCOM/ ($include 'IOCOM.INC')
929   C       INASCI = File unit of PREINFIL input control file
930   C       ISCREN = "            terminal screen
931   C       NOUTFL = "            diagnostics/debug file
932   C
933   C   LOCAL/
934   C       IL,IR  = Beg. & Ending positions of non-blank characters
935   C       IFILE  = Current file information being echoed to
936   C       IOSTAT = ANSI FORTRAN I/O status integer
937   C
938   C   EXIT/
939   C       none
940   C
941   C*********************************************************************
942   C234567
943         IMPLICIT NONE
944         INCLUDE 'IOCOM.INC/LIST'
945         INTEGER INFILUNT, OUTFILUNT, I, IERR, INREC, IOSTAT, IREC,
946        +        LASTREC
947         CHARACTER*(132) RECORD
948
949   C<><><><><><><><><><><><>
950   C...BEGIN PROCEDURES...
951   C<><><><><><><><><><><><>
952
953   C     ...Set initial values
954         IERR   = 0
955   C     ...Read input file
956    100  READ (INFILUNT,'(A)',IOSTAT=IOSTAT) RECORD
957
958         IF(IOSTAT.LT.0)THEN
959   C        ...E-O-F read
960            GOTO 999
```

```
961
962          ELSEIF(IOSTAT.GT.0)THEN
963   C         ...Error in while reading
964             IERR = IERR + 1
965   C         ...Read next record
966             GOTO 999
967
968          ELSEIF(IOSTAT.EQ.0) THEN
969             LASTREC = LASTREC + 1
970             IF(LASTREC .EQ. INREC) THEN
971   C           ...write to output file
972               IF (OUTFILUNT .GT. 0) WRITE (OUTFILUNT,'(A)') RECORD
973               GOTO 999
974             ELSE
975               IREC = IREC + 1
976               GOTO 100
977             ENDIF
978          ENDIF
979   C
980    999   IF(IERR.GT.0)THEN
981             WRITE(NOUTFL,*)' *** ',IERR,' Errors found in ECHOREC ***'
982             CALL QAABORT('ECHOREC')
983          ENDIF
984          RETURN
985   C-----------------------------------------------------------------------
986   C**** END OF SUBROUTINE ECHOREC ****
987   C-----------------------------------------------------------------------
988          END
```

```
      SUBROUTINE GTVECT(FILENMIN,FILENMOU)
C*****************************************************************
C  AUTHOR  Ron McCurley
C*****************************************************************
      INTEGER ICHAR
      CHARACTER*(*) FILENMIN, FILENMOU
      CHARACTER*(3) SAVSTRG

C...Find characters preceding vector index in filename, i.e. '_R'
      DO 50 ICHAR=1,80
         IF (FILENMIN(ICHAR:ICHAR+1).EQ.'_R')
     +       SAVSTRG=FILENMIN(ICHAR+2:ICHAR+4)
   50 CONTINUE
C,..  replace '###' string in output filename with 3 digit character index
      DO 100 ICHAR=1,20
         IF (FILENMOU(ICHAR:ICHAR+2).EQ. '###')
     +       FILENMOU(ICHAR:ICHAR+2) = SAVSTRG
  100 CONTINUE
      RETURN
      END

*DECK MEMADJ
      SUBROUTINE MEMADJ( MORMEM, MNUSED, Q )
C*********************************************************************
C***                                                               ****
C***              A_D_J_usts  dynamic  M_E_M_ory                   ****
C***             - - -                - - -                        ****
C*********************************************************************
C
C  PURPOSE:     Adjusts the amount of dynamic memory allocated
C               the to REAL base array, Q.
C
C  AUTHOR:      Jonathan S. Rath
C
C  UPDATED:     13-JUN-1994  --JSR: First Edition
C
C  CALLED BY:   DYNMEM
C               DYNMEM2 (entry point)
C               DYNMEM3 (entry point)
C
C  ENTRY POINT: MEMADJ2
C               MEMADJ3
C
C  CALLS:       (MEM_MGR):  MDGET       (V_EXT_LIB): EXEMEM
C                           MDRSRV
C                           MDLONG
C                           MDLIST
C                           MDSTAT
C                           MDEROR
C                           MDDEBG
C                           MDDEL
C                           MDCOMP
C                           MDGIV
C
C  ARGUMENTS:
C  ENTRY/
```

```
1046  C     --common blocks
1047  C       /IOCOM/ (from FORTRAN statement: 'IOCOM.INC')
1048  C        NOUTFL = File unit of diagnostics/debug output file
1049  C
1050  C       /LENGTH/ (from FORTRAN statement: 'LENGTH.INC')
1051  C        RLENGT = Dynamic REAL arrays LENGTHS array
1052  C
1053  C       /DYNAME/ (from FORTRAN statement: 'DYNAME.INC')
1054  C        RNAMES = Dynamic REAL arrays NAMES array
1055  C
1056  C       /INDEX/ (from FORTRAN statement: 'INDEX.INC')
1057  C        BRDAIDX = Beg. Real Dynamic Array InDeX for memory allocation
1058  C        ERDAIDX = End. Real "
1059  C
1060  C       /MISCDAT/ (from FORTRAN statement: 'MISCDAT.INC')
1061  C        DEBUG  = Logical debugging tool
1062  C
1063  C     --through subroutine call
1064  C        MORMEM = Add'l no. of words needed for core allocation (REAL-Q)
1065  C        Q      = Dynamic Memory base REAL array
1066  C
1067  C  LOCAL/
1068  C        MNERRS =  Number of Memory manager ERRorS during execution
1069  C
1070  C  EXIT/
1071  C     --common blocks
1072  C       /LOCAT/ (from FORTRAN statement: 'LOCAT.INC')
1073  C        LRPTR  = Pointer array used for REAL dynamic array (Q)
1074  C
1075  C     --through subroutine call
1076  C        MNUSED = Total words allocated to REAL dynamic array
1077  C        Q      = Dynamic Memory base REAL array
1078  C
1079  C*********************************************************************
1080  C234567
1081        IMPLICIT NONE
1082        INCLUDE 'IOCOM.INC'
1083        INCLUDE 'LOCAT.INC'
1084        INCLUDE 'DYNAME.INC'
1085        INCLUDE 'LENGTH.INC'
1086        INCLUDE 'INDEX.INC'
1087        INCLUDE 'MISCDAT.INC'
1088        INTEGER I, I1, I2, MORMEM, MNERRS, MNUSED
1089        REAL Q(*)
1090  C
1091  C*********************************************************************
1092  C
1093  C     PSEUDO ENTRY MEMADJ( MORMEM, MNUSED, Q )
1094  C
1095  C*********************************************************************
1096  C
1097  C     THIS SECTION SETS DYNAMIC ARRAY LENGTHS FOR FIRST TIME
1098  C
1099  C     ...Check for bad REAL dynamic array indices
1100        IF(BRDAIDX.LT.1.OR.ERDAIDX.GT.LASRIDX.OR.
1101     +    ERDAIDX.LT.1.OR.BRDAIDX.GT.LASRIDX    )THEN
1102          WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
```

```
1103            WRITE(NOUTFL,*)'     BRDAIDX = ',BRDAIDX
1104            WRITE(NOUTFL,*)'     ERDAIDX = ',ERDAIDX
1105            WRITE(NOUTFL,*)'     LASRIDX = ',LASRIDX
1106            CALL QAABORT('(MEMADJ) - Invalid REAL dynamic array '//
1107         +              'indices detected ***')
1108         ENDIF
1109         CALL MDGET(MORMEM)
1110         CALL MDFILL(1.0)
1111         DO 100 I = BRDAIDX, ERDAIDX, 1
1112            CALL MDRSRV(RNAMES(I),I1,RLENGT(I))
1113            LRPTR(I) = I1
1114   100   CONTINUE
1115 C       ...Check for dynamic memory allocation errors
1116         IF(DEBUG)CALL MDDEBG(NOUTFL)
1117         CALL MDSTAT(MNERRS,MNUSED)
1118         IF(MNERRS.NE.0.OR.DEBUG)CALL MDLIST(NOUTFL)
1119         IF(MNERRS.NE.0.        )THEN
1120            CALL MDEROR(NOUTFL)
1121            CALL QAABORT('(MEMADJ) - REAL DYNAMIC MEMORY '//
1122         +              'ALLOCATION ERRORS ***')
1123         ENDIF
1124         RETURN
1125 C
1126 C*********************************************************************
1127 C
1128         ENTRY MEMADJ2( MORMEM, MNUSED, Q )
1129 C
1130 C*********************************************************************
1131 C
1132 C       THIS SECTION LENGTHENS OR SHORTENS DYNAMIC ARRAYS
1133 C
1134 C       ...Check for bad REAL dynamic array indices
1135         IF(BRDAIDX.LT.1.OR.ERDAIDX.GT.LASRIDX.OR.
1136         +  ERDAIDX.LT.1.OR.BRDAIDX.GT.LASRIDX    )THEN
1137            WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
1138            WRITE(NOUTFL,*)'     BRDAIDX = ',BRDAIDX
1139            WRITE(NOUTFL,*)'     ERDAIDX = ',ERDAIDX
1140            WRITE(NOUTFL,*)'     LASRIDX = ',LASRIDX
1141            CALL QAABORT('(MEMADJ2) - Invalid REAL dynamic array '//
1142         +              'indices detected ***')
1143         ENDIF
1144         DO 200 I = BRDAIDX, ERDAIDX, 1
1145            CALL MDLONG(RNAMES(I),I1,RLENGT(I))
1146            LRPTR(I) = I1
1147   200   CONTINUE
1148 C       ...Check for dynamic memory allocation errors
1149         IF(DEBUG)CALL MDDEBG(NOUTFL)
1150         CALL MDSTAT(MNERRS,MNUSED)
1151         IF(MNERRS.NE.0.OR.DEBUG)CALL MDLIST(NOUTFL)
1152         IF(MNERRS.NE.0.        )THEN
1153            CALL MDEROR(NOUTFL)
1154            CALL QAABORT('(MEMADJ2) - REAL '//
1155         $                'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1156         ENDIF
1157         RETURN
1158 C
1159 C*********************************************************************
```

```
1160  C
1161        ENTRY MEMADJ3( MNUSED, Q )
1162  C
1163  C*********************************************************************
1164  C
1165  C     THIS SECTION CONSOLIDATES FRAGMENTED MEMORY, RETURNS UNUSED
1166  C     STORAGE TO THE OPERATING SYSTEM, AND THEN RECALCULATES ALL
1167  C     DYNAMIC ARRAY POINTERS.
1168  C
1169  C
1170  C     ...Consolidate the fragmented memeory
1171        CALL MDCOMP
1172  C     ...Return unused storage back to the operating system
1173        CALL MDGIVE
1174  C     ...Recalculate ALL pointers of the Q-array
1175        DO 300 I = BRDAIDX, ERDAIDX, 1
1176           I2 = 0
1177           CALL MDFIND(RNAMES(I),I1,I2)
1178           LRPTR(I)  = I1
1179           RLENGT(I) = I2
1180   300  CONTINUE
1181  C     ...Check for dynamic memeory allocation errors
1182        IF(DEBUG)CALL MDDEBG(NOUTFL)
1183        CALL MDSTAT(MNERRS,MNUSED)
1184        IF(MNERRS.NE.0.OR.DEBUG)CALL MDLIST(NOUTFL)
1185        IF(MNERRS.NE.0.        )THEN
1186           CALL MDEROR(NOUTFL)
1187           CALL QAABORT('(MEMADJ3) - REAL '//
1188       $                 'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1189        ENDIF
1190        RETURN
1191  C---------------------------------------------------------------------
1192  C**** END OF SUBROUTINE MEMADJ ****
1193  C---------------------------------------------------------------------
1194        END
```

```
1195
1196  *DECK MEMADJC
1197        SUBROUTINE MEMADJC( MORMEMC, MNUSED, C )
1198  C*********************************************************************
1199  C***                                                               ****
1200  C***       A_D_J_usts  dynamic  M_E_M_ory  (C_haracter type)       ****
1201  C***       - - -                - - -       -                      ****
1202  C*********************************************************************
1203  C
1204  C  PURPOSE:     Adjusts the amount of dynamic memory allocated to
1205  C               the CHARACTER*1 base array, C.
1206  C
1207  C  AUTHOR:      Jonathan S. Rath
1208  C
1209  C  UPDATED:     13-JUN-1994  --JSR: First Edition
1210  C
1211  C  CALLED BY:   DYNMEMC
1212  C               DYNMEMC2 (entry point)
1213  C               DYNMEMC3 (entry point)
1214  C
1215  C  ENTRY POINT: MEMADJC2
1216  C               MEMADJC3
1217  C
1218  C  CALLS:       (MEM_MGR):  MCGET       (V_EXT_LIB): EXEMEM
1219  C                           MCRSRV
1220  C                           MCLONG
1221  C                           MCLIST
1222  C                           MCSTAT
1223  C                           MCEROR
1224  C                           MCFIL
1225  C                           MCDEBG
1226  C                           MCCOMP
1227  C                           MCGIV
1228  C
1229  C  ARGUMENTS:
1230  C  ENTRY/
1231  C    --common blocks
1232  C      /IOCOM/ (from FORTRAN statement: 'IOCOM.INC')
1233  C       NOUTFL = File unit of diagnostics/debug output file
1234  C
1235  C      /LENGTH/ (from FORTRAN statement: 'LENGTH.INC')
1236  C       CLENGT = Dynamic CHAR arrays LENGTHS array
1237  C
1238  C      /DYNAME/ (from FORTRAN statement: 'DYNAME.INC')
1239  C       CNAMES = Dynamic CHAR arrays NAMES array
1240  C
1241  C      /INDEX/ (from FORTRAN statement: 'INDEX.INC')
1242  C       BCDAIDX = Beg. Char Dynamic Array InDeX for memory allocation
1243  C       ECDAIDX = End. Char "
1244  C
1245  C      /MISCDAT/ (from FORTRAN statement: 'MISCDAT.INC')
1246  C       DEBUG  = Logical debugging tool
1247  C
1248  C     --through subroutine call
1249  C       MORMEMC = Add'l no. of words needed for core allocation (CHAR-C)
1250  C       C       = Dynamic memory CHARACTER*1 base array
1251  C
```

```
1252  C     --through ENTRY POINT
1253  C       NAME   = Name of array to be deleted
1254  C
1255  C  LOCAL/
1256  C       MNERRS =  Number of Memory manager ERRorS during execution
1257  C
1258  C  EXIT/
1259  C     --common blocks
1260  C      /LOCAT/ (from FORTRAN statement: 'LOCAT.INC')
1261  C       LCPTR  = Pointer array used for CHAR dynamic array (C)
1262  C
1263  C     --through subroutine call
1264  C       MNUSED = Total words allocated to CHARACTER dynamic array
1265  C       C      = Dynamic Memory CHARACTER*1 base array
1266  C
1267  C*********************************************************************
1268  C234567
1269        IMPLICIT NONE
1270        INCLUDE 'IOCOM.INC'
1271        INCLUDE 'LOCAT.INC'
1272        INCLUDE 'DYNAME.INC'
1273        INCLUDE 'LENGTH.INC'
1274        INCLUDE 'INDEX.INC'
1275        INCLUDE 'MISCDAT.INC'
1276        INTEGER I, I1, I2, MORMEMC, MNERRS, MNUSED
1277        CHARACTER*(1) C(*)
1278  C
1279  C*********************************************************************
1280  C
1281  C     PSEUDO ENTRY MEMADJC( MORMEMC, MNUSED, C )
1282  C
1283  C*********************************************************************
1284  C
1285  C     THIS SECTION SETS DYNAMIC ARRAY LENGTHS FOR FIRST TIME
1286  C
1287  C     ...Check for bad CHARACTER dynamic array indices
1288        IF(BCDAIDX.LT.1.OR.ECDAIDX.GT.LASCIDX.OR.
1289     +    ECDAIDX.LT.1.OR.BCDAIDX.GT.LASCIDX    )THEN
1290           WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
1291           WRITE(NOUTFL,*)'     BCDAIDX = ',BCDAIDX
1292           WRITE(NOUTFL,*)'     ECADIDX = ',ECDAIDX
1293           WRITE(NOUTFL,*)'     LASCIDX = ',LASCIDX
1294           CALL QAABORT('(MEMADJ) - Invalid CHARACTER dynamic array '//
1295     +              'indices detected ***')
1296        ENDIF
1297        CALL MCGET(MORMEMC)
1298        CALL MCFILL(' ')
1299        DO 100 I = BCDAIDX, ECDAIDX, 1
1300           CALL MCRSRV(CNAMES(I),I1,CLENGT(I))
1301           LCPTR(I) = I1
1302   100  CONTINUE
1303  C     ...Check for dynamic memory allocation errors
1304        IF(DEBUG)CALL MCDEBG(NOUTFL)
1305        CALL MCSTAT(MNERRS,MNUSED)
1306        IF(MNERRS.NE.0.OR.DEBUG)CALL MCLIST(NOUTFL)
1307        IF(MNERRS.NE.0.        )THEN
1308           CALL MCEROR(NOUTFL)
```

```
1309            CALL QAABORT('(MEMADJC) - CHARACTER DYNAMIC MEMORY '//
1310         +                'ALLOCATION ERRORS ***')
1311          ENDIF
1312          RETURN
1313   C
1314   C*********************************************************************
1315   C
1316          ENTRY MEMADJC2( MORMEMC, MNUSED, C )
1317   C
1318   C*********************************************************************
1319   C
1320   C      THIS SECTION LENGTHENS OR SHORTENS DYNAMIC ARRAYS
1321   C
1322   C      ...Check for bad CHARACTER dynamic array indices
1323          IF(BCDAIDX.LT.1 .OR. ECDAIDX.GT.LASCIDX .OR.
1324         +   ECDAIDX.LT.1 .OR. BCDAIDX.GT.LASCIDX    )THEN
1325            WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
1326            WRITE(NOUTFL,*)'     BCDAIDX = ',BCDAIDX
1327            WRITE(NOUTFL,*)'     ECADIDX = ',ECDAIDX
1328            WRITE(NOUTFL,*)'     LASCIDX = ',LASCIDX
1329            CALL QAABORT('(MEMADJC2) - Invalid CHARACTER dynamic array '//
1330         +                'indices detected ***')
1331          ENDIF
1332          DO 200 I = BCDAIDX, ECDAIDX, 1
1333            CALL MCLONG(CNAMES(I),I1,CLENGT(I))
1334            LCPTR(I) = I1
1335    200   CONTINUE
1336          CALL MCFILL(' ')
1337   C      ...Check for dynamic memory allocation errors
1338          IF(DEBUG)CALL MCDEBG(NOUTFL)
1339          CALL MCSTAT(MNERRS,MNUSED)
1340          IF(MNERRS.NE.0.OR.DEBUG)CALL MCLIST(NOUTFL)
1341          IF(MNERRS.NE.0.         )THEN
1342            CALL MCEROR(NOUTFL)
1343            CALL QAABORT('(MEMADJC2) - CHARACTER '//
1344         $                'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1345          ENDIF
1346          RETURN
1347   C
1348   C*********************************************************************
1349   C
1350          ENTRY MEMADJC3( MNUSED, C )
1351   C
1352   C*********************************************************************
1353   C
1354   C      THIS SECTION CONSOLIDATES FRAGMENTED MEMORY, RETURNS UNUSED
1355   C      STORAGE TO THE OPERATING SYSTEM, AND THEN RECALCULATES ALL
1356   C      DYNAMIC ARRAY POINTERS.
1357   C
1358   C      ...Consolidate the fragmented memory
1359          CALL MCCOMP
1360   C      ...Return unused storage back to the operating system
1361          CALL MCGIVE
1362   C      ...Recalculate ALL pointers of the C-array
1363          DO 300 I = BCDAIDX, ECDAIDX, 1
1364          I2 = 0
1365          CALL MCFIND(CNAMES(I),I1,I2)
```

```
1366            LCPTR(I)  = I1
1367            CLENGT(I) = I2
1368     300  CONTINUE
1369   C      ...Check for dynamic memeory allocation errors
1370          IF(DEBUG)CALL MCDEBG(NOUTFL)
1371          CALL MCSTAT(MNERRS,MNUSED)
1372          IF(MNERRS.NE.0.OR.DEBUG)CALL MCLIST(NOUTFL)
1373          IF(MNERRS.NE.0.        )THEN
1374             CALL MCEROR(NOUTFL)
1375             CALL QAABORT('(MEMADJC3) - ### CHARACTER ### '//
1376         $                'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1377          ENDIF
1378          RETURN
1379   C-----------------------------------------------------------------------
1380   C**** END OF SUBROUTINE MEMADJC ****
1381   C-----------------------------------------------------------------------
1382          END
```

```
1383
1384  *DECK,OUTPUT
1385        SUBROUTINE OUTPUT
1386  C***********************************************************************
1387  C
1388  C  PURPOSE:        Writes ASCII information to file NOUTFL about
1389  C                  the LHS input translation run.
1390  C
1391  C  AUTHORS:        Ron McCurley, Jonathan S. Rath
1392  C
1393  C  UPDATED:        16 January, 1989  --J.S.R. adapted to new version
1394  C                  23 May, 1989 --J.S.R. deleted CLOSE statements
1395  C                  19 May, 1993 --L.N.S. replaced all ECHO2 calls with
1396  C                                        calls to subroutine FILECHO2.
1397  C
1398  C  CALLED BY:      PRELHS
1399  C
1400  C  CALLS:          FILECHO2
1401  C
1402  C  ARGUMENTS:
1403  C  ENTRY/
1404  C     --common block
1405  C      /IOCOM/ ($include 'IOCOM.INC')
1406  C       NOUTFL = File unit of diagnostics/debug output file
1407  C
1408  C  EXIT/
1409  C       none
1410  C
1411  C***********************************************************************
1412  C234567
1413        IMPLICIT NONE
1414        INCLUDE 'IOCOM.INC/LIST'
1415
1416        WRITE(NOUTFL,*)' This is the PREINFIL input information ECHOED'
1417        RETURN
1418  C-----------------------------------------------------------------------
1419  C**** END OF SUBROUTINE OUTPUT *****
1420  C-----------------------------------------------------------------------
1421        END
```

```
1422
1423  *DECK,PREPRO
1424        SUBROUTINE PREPRO( FUSR, FCDB, FTMP, FINF, FOUT )
1425  C*********************************************************************
1426  C
1427  C  PURPOSE:     PREPRO input preprocessor which performs:
1428  C               1.  QA functions
1429  C               2.  Calls routine for parsing command line argUements
1430  C                   which consists of filenames.  Those files are
1431  C                   opened when appropriate or prompted for if not
1432  C                   supplied on the command line or set to defaults.
1433  C
1434  C  AUTHOR:      Ron D McCurley
1435  C
1436  C  UPDATED:     07 May, 1999  --
1437  C
1438  C  CALLED BY:   PREINFIL {main program}
1439  C
1440  C  CALLS:       EXPARM
1441  C               FILCMDLIN
1442  C               WELCOM
1443  C               TYPRQS
1444  C               ISTRLEN
1445  C               QABANNER
1446  C               QAPAGE
1447  C               QADOEDIS
1448  C
1449  C  ARGUMENTS:
1450  C  ENTRY/
1451  C   --common blocks
1452  C   /IOCOM/ ($include 'IOCOM.INC')
1453  C    IOUTFL  = Device no. of diagnostics/debUg output file
1454  C              (Equilabrated to NOUTFL)
1455  C   /L1/  Contains Line 1 of a 3-line program discription written out
1456  C         following the program banner
1457  C   /L2/  Contains Line 2 of a 3-line program discription written out
1458  C         following the program banner
1459  C   /L3/  Contains Line 3 of a 3-line program discription written out
1460  C         following the program banner
1461  C
1462  C  LOCAL/
1463  C     NFILES = Maximum nUmber of files on command line
1464  C     BATCHF = Logical BATCH process flag
1465  C     INTRAF = Logical INTERACTIVE process flag
1466  C     ERRORF = Logical ERROR flag
1467  C     HARD   = System hardware ID
1468  C     SOFT   = System software ID
1469  C     MODE   = BATCH(0) or INTERACTIVE(1) mode
1470  C     KCSU   = Characters Units per base Unit
1471  C     KNSU   = Numeric storage Units per base Unit
1472  C     IDAU   = Units of storage which  define size of Unformatted
1473  C              direct I/O  records 0=character, 1=nUmeric
1474  C
1475  C  EXIT/
1476  C   --common block
1477  C   /IOCOM/ ($include 'IOCOM.INC')
1478  C    NOUTFL  = Device no. of diagnostics/debug output file
```

```
1479  C
1480  C   --through subroutine call
1481  C     FUSR     = PRE-INFIL Control Card Data filename
1482  C     FCDB     = Calculational data base filename
1483  C     FTMP     = Template INFIL input file
1484  C     FINF     = PREINF transfer file
1485  C     FOUT     = PREINF program diagnostics/debug filename
1486  C
1487  C-----------------------------------------------------------------------
1488  C234567
1489        IMPLICIT NONE
1490        INTEGER  ISTRLEN
1491        INCLUDE 'IOCOM.INC/LIST'
1492        COMMON /L1/  LINE1
1493        COMMON /L2/  LINE2
1494        COMMON /L3/  LINE3
1495
1496        INTEGER        I,      IDAU,    IOUTFL, KCSU,  KNSU, MODE, NFILES
1497        LOGICAL        EXIST, BATCHF, ERRORF, INTRAF
1498        CHARACTER*8    HARD,   SOFT
1499        CHARACTER*(*) FUSR,   FTMP, FINF, FOUT, FCDB
1500        CHARACTER*80  FILESP(5)
1501        CHARACTER*80  LINE1,LINE2,LINE3
1502
1503  C<><><><><><><><><><><><>
1504  C...BEGIN PROCEDURES...
1505  C<><><><><><><><><><><><>
1506
1507  C     ...Check if current run is BATCH or INTERACTIVE
1508  C        BATCH -> messages to debug file
1509  C        INTER -> messages to screen
1510        CALL EXPARM(HARD,SOFT,MODE,KCSU,KNSU,IDAU)
1511        IF (MODE .EQ. 0) THEN
1512           BATCHF = .TRUE.
1513        ELSE
1514           BATCHF = .FALSE.
1515        ENDIF
1516
1517  C     ...Get files names from command line
1518  C        FILESP(1 to 5) are: FUSR, FCDB, FTMP, FINF, FOUT
1519  C     ...If FUSR = 'default' all files will be defaulted, otherwise
1520  C        they will be prompted for.  A blank response to the prompt will
1521  C        also result in default file name.
1522        ERRORF = .FALSE.
1523        CALL FILCMDLIN(5,NFILES,FILESP)
1524        FUSR = FILESP(1)
1525        FCDB = FILESP(2)
1526        FTMP = FILESP(3)
1527        FINF = FILESP(4)
1528        FOUT = FILESP(5)
1529
1530  C     ...If any files specified on command line set INTRAF to false
1531        INTRAF = .TRUE.
1532        IF (NFILES .GT. 0) THEN
1533           INTRAF = .FALSE.
1534           BATCHF = .TRUE.
1535        ELSE
```

```
1536            INTRAF = .TRUE.
1537         ENDIF
1538
1539         IF (INTRAF) THEN
1540   C     ...INTERACTIVE:  Prompt for filenames
1541
1542            CALL WELCOM
1543            WRITE(*,1000)
1544   C
1545   C        ------------------------------
1546   C        PRE-INFIL Control Card Data file
1547   C        ------------------------------
1548
1549     100    CALL TYPRQS(' Enter PREINF Control Card Data filename'//
1550        +                ' <PREINF.INP>',FUSR)
1551   C        ...Null response implies default
1552            IF (FUSR .EQ. 'default' .OR. FUSR .EQ. 'DEFAULT' .OR.
1553        &       FUSR .EQ. ' ') FUSR = 'PREINF.INP'
1554            INQUIRE(FILE=FUSR, EXIST=EXIST)
1555            IF (.NOT. EXIST) THEN
1556               WRITE(*,'(3A/)')
1557        &        ' FUSR=',FUSR(1:ISTRLEN(FUSR)),' does not exist'
1558               FUSR = ' '
1559               GOTO 100
1560            ENDIF
1561
1562   C        -------------------
1563   C        COMPUTATIONAL data base
1564   C        -------------------
1565   C
1566   C        ...Prompt User for COMPUTATIONAL data base
1567    200     CALL TYPRQS(' Enter COMPUTATIONAL data base filename'//
1568        +                ' <POSTLHS.CDB>',FCDB)
1569   C        ...Null response implies default
1570            IF (FCDB .EQ. 'default' .OR. FCDB .EQ. 'DEFAULT' .OR.
1571        &       FCDB .EQ. ' ') FCDB='POSTLHS.CDB'
1572            INQUIRE(FILE=FCDB, EXIST=EXIST)
1573            IF (.NOT. EXIST) THEN
1574               WRITE(*,'(3A/)')
1575        &        ' FCDB=',FCDB(1:ISTRLEN(FCDB)),' does not exist'
1576               FCDB = ' '
1577               GOTO 200
1578            ENDIF
1579
1580   C        ---------------------
1581   C        PREINFIL input INFIL template control file
1582   C        ---------------------
1583   C
1584   C        ...Prompt User for PREINFIL input INFIL template filename
1585            CALL TYPRQS(' Enter PREINFIL input INFIL template filename'//
1586        +                ' <INFIL.TMP>',FTMP)
1587   C        ...Null response implies default
1588            IF (FTMP .EQ. 'default' .OR. FTMP .EQ. 'DEFAULT' .OR.
1589        &       FTMP .EQ. ' ') FTMP = 'INFIL.TMP'
1590
1591   C        ---------------------
1592   C        PREINFIL output transfer file (to INFIL) input control file
```

```
1593  C         ---------------------
1594  C
1595  C         ...Prompt User for PREINFIL transfer filename
1596            CALL TYPRQS(' Enter PREINF to INFIL transfer filename'//
1597       +                ' <INFIL.CTL>',FINF)
1598  C         ...Null response implies default
1599            IF (FINF .EQ. 'default' .OR. FINF .EQ. 'DEFAULT' .OR.
1600       &       FINF .EQ. ' ') FINF = 'INFIL.CTL'
1601
1602  C         ----------------------------
1603  C         Diagnostics/Debug output file
1604  C         ----------------------------
1605  C
1606            WRITE(*,1100)
1607  C         ...Prompt User for OPTIONAL diagnostics/debUg file
1608            CALL TYPRQS(' Enter (optional) PREINF diagnostics/debug'//
1609       +                ' filename <PREINF.DBG>',FOUT)
1610  C         ...Null response implies default
1611            IF (FOUT .EQ. 'default' .OR. FOUT .EQ. 'DEFAULT' .OR.
1612       &       FOUT .EQ. ' ') FOUT='PREINF.DBG'
1613            IF (FOUT(:3) .EQ. 'CAN' .OR. FOUT(:3) .EQ. 'can') THEN
1614  C            ...Don't write a recoverable diagnostics/debug file
1615               NOUTFL = 6
1616               FOUT   = 'PREINF.SCR'
1617            ELSE
1618               NOUTFL = 7
1619            ENDIF
1620
1621         ELSE
1622
1623  C      ...Set Undefined files to defaults and check for existence
1624            IF (FUSR .EQ. 'default' .OR. FUSR .EQ. 'DEFAULT' .OR.
1625       &     (BATCHF .AND. FUSR .EQ. ' ') ) FUSR = 'PRECTL.INP'
1626            INQUIRE(FILE=FUSR,EXIST=EXIST)
1627            IF (.NOT.EXIST) THEN
1628               WRITE(*,'(3A/)')
1629       &         ' FUSR=',FUSR(1:ISTRLEN(FUSR)),' does not exist'
1630               ERRORF = .TRUE.
1631            ENDIF
1632
1633            IF (FCDB .EQ. 'DEFAULT' .OR. FCDB .EQ. 'default' .OR.
1634       &     (BATCHF .AND. FCDB .EQ. ' ') ) FCDB = 'POSTLHS.CDB'
1635            INQUIRE(FILE=FCDB,EXIST=EXIST)
1636            IF (.NOT.EXIST) THEN
1637               WRITE(*,'(3A/)')
1638       &         ' FCDB=',FCDB(1:ISTRLEN(FCDB)),' does not exist'
1639               ERRORF = .TRUE.
1640            ENDIF
1641
1642            IF (FTMP .EQ. 'default' .OR. FTMP .EQ. 'DEFAULT' .OR.
1643       &     (BATCHF .AND. FTMP .EQ. ' ') ) FTMP = 'INFIL.TMP'
1644
1645            IF (FINF .EQ. 'default' .OR. FINF .EQ. 'DEFAULT' .OR.
1646       &     (BATCHF .AND. FINF .EQ. ' ') ) FINF = 'INFIL.CTL'
1647
1648            IF (FOUT .EQ. 'default' .OR. FOUT .EQ. 'DEFAULT' .OR.
1649       &     (BATCHF .AND. FOUT .EQ. ' ') ) FOUT = 'PREINFIL.DBG'
```

```
1650
1651            IF (FOUT(:3).EQ. 'CAN' .OR. FOUT(:3).EQ. 'can') THEN
1652   C ...Don't write a recoverable diagnostics/debug file
1653               NOUTFL = 6
1654               FOUT   = 'PREINFIL.SCR'
1655            ELSE
1656               NOUTFL = 7
1657            ENDIF
1658         ENDIF
1659
1660   C     ...Open Diagnostics/Debug output file
1661         IF (NOUTFL.EQ.7) THEN
1662            OPEN(NOUTFL,FILE=FOUT,FORM='FORMATTED',STATUS='UNKNOWN')
1663         ELSEIF (NOUTFL.EQ.6) THEN
1664            OPEN(NOUTFL,FILE=FOUT,STATUS='SCRATCH')
1665         ENDIF
1666
1667   C     ...Write QA stuff
1668         CALL QABANNER(NOUTFL,LINE1,LINE2,LINE3)
1669         CALL QAPAGE(NOUTFL,' ')
1670         CALL QADOEDIS(NOUTFL,'*')
1671
1672         WRITE(NOUTFL,1200)
1673         WRITE(NOUTFL,'(A)')'            FILE ASSIGNMENTS'
1674         WRITE(NOUTFL,'(A)')'            ----------------'
1675         WRITE(NOUTFL,'(A,A)')' PREINFIL Input Control FILE...........',
1676        +                     FUSR(1:40)
1677         WRITE(NOUTFL,'(A,A)')' COMPUTATIONAL Data Base FILE............',
1678        +                     FCDB(1:40)
1679         WRITE(NOUTFL,'(A,A)')' PREINFIL input INFIL Template FILE......',
1680        +                     FTMP(1:40)
1681         WRITE(NOUTFL,'(A,A)')' PREINFIL to INFIL Transfer FILE.........',
1682        +                     FINF(1:40)
1683         WRITE(NOUTFL,'(A,A)')' DIAGNOSTICS/DEBUG Output FILE.......',
1684        +                     FOUT(1:40)
1685
1686   C     ...Equate IOUTFL to NOUTFL
1687         IOUTFL = NOUTFL
1688
1689   C     ...Stop if ERRORF
1690         IF (ERRORF) THEN
1691            WRITE(NOUTFL,'(A)')' FILE SPECIFICATION ERROR(S)'
1692            STOP '*** FILE SPECIFICATION ERROR(S) IN PREPRO ***'
1693         ENDIF
1694         RETURN
1695   C
1696   C------------------ F O R M A T   S T A T E M E N T S ------------------
1697   C
1698   1000  FORMAT(/,' ...For <default filenames>, press RETURN',/)
1699   1100  FORMAT(/,' ...To CANCEL optional files requested, respond',
1700        +         ' with: CANCEL',/)
1701   1200  FORMAT(/,79('*'),/,1X,'(PREPRO)')
1702   C
1703   C-----------------------------------------------------------------------
1704   C**** END OF subroutine PREPRO ****
1705   C-----------------------------------------------------------------------
1706         END
```

```
1707
1708  *DECK PROCDB
1709        SUBROUTINE PROCDB( MAXC, MAXQ, C, IQ, LQ, Q )
1710  C*********************************************************************
1711  C****                                                             ****
1712  C****          P_R_O_C_ess  input D_ata  B_ase  routine           ****
1713  C****          - - - -            -      -                        ****
1714  C*********************************************************************
1715  C
1716  C  PURPOSE:    Process dynamic memory allocation for input
1717  C              CAMDAT file etc...
1718  C
1719  C  AUTHOR:     Jonathan S. Rath
1720  C
1721  C  UPDATED:    18-MAY-1999  --(RDM) MODIFIED FOR PREINFIL
1722  C              16-APR-1992  --Added arguments in REAHEA
1723  C                           --Added calls to ZERODA
1724  C              27-MAR-1992  --Changed arguments in GIEBLO
1725  C              17-DEC-1991  --First Ed.
1726  C
1727  C  CALLED BY:  PREINFIL{main program}
1728  C
1729  C  CALLS:      RECHEA
1730  C              REATIM
1731  C
1732  C  ARGUMENTS:
1733  C  ENTRY/
1734  C   --common blocks
1735  C    /IOCOM/ ($INCLUDE 'IOCOM.INC')
1736  C     ICAMBFN    = Device no of Input CAMDAT File
1737  C     OCAMBFN    = "            Output CAMDAT file
1738  C
1739  C   --through subroutine call
1740  C     MAXC    = Total no. of CHAR*1 words allocated to dynamic storage
1741  C     MAXQ    = Total no. of words allocated to dynamic storage
1742  C     IQ      = INTEGER Dynamic Q-array (equivalenced to Q)
1743  C     LQ      = LOGICAL Dynamic Q-array (equivalenced to Q)
1744  C     Q       = REAL Dynamic memory base array
1745  C     C       = CHARACTER Dynamic memory base array
1746  C
1747  C  LOCAL/
1748  C     IERR    = INTEGER error code
1749  C     MORMEMC= No. additional words to be allocated to C array
1750  C     MORMEMR= No. additoinal words to be allocated to Q array
1751  C
1752  C  EXIT/
1753  C   --common blocks
1754  C    /LOCATR/ ($INCLUDE 'LOCAT.INC')
1755  C     L[name]= Pointer to [name] Q-array integers
1756  C     LRPTR  = Pointer of Q-array
1757  C
1758  C    /LOCATC/ ($INCLUDE 'LOCAT.INC')
1759  C     L[name]= Pointer to [name] C-array integers
1760  C     LCPTR  = Pointer of C-array
1761  C
1762  C    /GEOPAR/ ($INCLUDE 'GEOPAR.INC')
1763  C     HEADER = Character description of problem, etc...
```

```
1764  C      IMAX   = No. of I-lines in grid
1765  C      JMAX   = No. of J-lines in grid
1766  C      KMAX   = No. of K-lines in grid
1767  C      LESSEL = List of Element Side sets concatenated Element List
1768  C      LESSNL = List of Element Side sets concatenated Nodes associated
1769  C               with each element side set Node List
1770  C      LNPSNL = List of Nodal Point set concatenated Nodes List
1771  C      MAXNENO= Max. no. of nodes of any element
1772  C      NDIM   = Number of physical DIMensions
1773  C      NDBCAM = CAMDAT versoin number
1774  C      NELBLK = Number of ELement BLocKs in mesh
1775  C      NELX   = No. of elements along X-axis of mesh
1776  C      NELY   = No. of elements along Y-axis of mesh
1777  C      NELZ   = No. of elements along Z-axis of mesh
1778  C      NQAREC = Number of QA records
1779  C      NUMEL  = NUmber of ELeMents in mesh
1780  C      NUMESS = NUMber of Element Side set in mesh
1781  C      NUMNOD = NUMber of NODes in mesh
1782  C      NUMNPS = NUMber of Nodal Point set in mesh
1783  C      NVAREL = No. of ELEMENT type ANALYSIS variables
1784  C      NVARGL = No. of GLOBAL  "                    "
1785  C      NVARHI = No. of HISTORY "                    "
1786  C      NVARND = No. of NODA    "                    "
1787  C
1788  C*********************************************************************
1789  C234567
1790        IMPLICIT NONE
1791        INCLUDE 'IOCOM.INC/LIST'
1792        INCLUDE 'LOCAT.INC/LIST'
1793        INCLUDE 'GEOPAR.INC/LIST'
1794        INTEGER  IERR, MAXC, MAXQ, MORMEMC, MORMEMR
1795        INTEGER IQ(*)
1796        REAL Q(*)
1797        CHARACTER*(1) C(*)
1798        LOGICAL LQ(*)
1799  C
1800  C<><><><><><><><><><><><>
1801  C...Begin Procedures...
1802  C<><><><><><><><><><><><>
1803  C
1804        IERR = 0
1805  C
1806        CALL DBISIZES( ICDB,     NQAREC,  NXINFO,  NDIM,   NUMNOD,
1807       $               NELBLK,  NUMEL,   NELX,    NELY,   NELZ,
1808       $               MAXNENO, NUQATR,  NUQPRP,  NUMNPS, LNPSNL,
1809       $               NUMESS,  LESSEL,  LESSNL,  IERR )
1810
1811        CALL ADDMEM(MORMEMR,MORMEMC)
1812        CALL MDDEBG(NOUTFL)
1813        CALL DYNMEM( MORMEMR, MAXQ, Q )
1814        CALL DYNMEMC( MORMEMC, MAXC, C )
1815
1816  C     ...Read various CAMDAT "header" sections into arrays
1817        CALL RECHEA( Q,    C,    IQ(LIDEBLK), IQ(LNUMPRP), LQ(LIASPRP),
1818       $             Q(LXMATPR), C(LQAINFO), C(LNAMELB), C(LNMATPR) )
1819  C     ...Read various CAMDAT "analysis/times" section into arrays
1820  C*      CALL REATIM( Q,          C,          C(LNAMHIV), C(LNAMGLV),
```

```
1821  C*     1              C(LNAMNDV), C(LNAMELV), Q(LTIME),   Q(LHIFLAG),
1822  C*     2              Q(LVALHIV), Q(LVALGLV), Q(LVALNDV), Q(LVALELV) )
1823        RETURN
1824  C-----------------------------------------------------------------------
1825  C**** END OF SUBROUTINE PROCDB *****
1826  C-----------------------------------------------------------------------
1827        END
```

```
1828
1829  *DECK,PROCTL
1830        SUBROUTINE PROCTL
1831  C*********************************************************************
1832  C***                                                               ****
1833  C***              P_R_Ocess  input  C_on_T_ro_L File  module       ****
1834  C***              - - -              -    -    -                  ****
1835  C*********************************************************************
1836  C
1837  C  PURPOSE :     Reads the PREINFIL input control file and RETURNS the
1838  C                variables which will be modified from the template file
1839  C                (i.e.--the sampled parameters).  May also return file
1840  C                names as used by INFIL
1841  C
1842  C  AUTHOR:       Ron D McCurley
1843  C
1844  C  UPDATED:      May, 1999
1845  C
1846  C  CALLED BY:    PREINFIL
1847  C
1848  C  CALLS:        FFRDFLDS
1849  C                RETRIE
1850  C                QAABORT
1851  C
1852  C  ARGUMENTS:
1853  C  ENTRY/
1854  C   --common blocks
1855  C    /IOCOM/ ($include 'IOCOM.INC')
1856  C     INASCI = File unit of PRELHS input control file
1857  C     NOUTFL = "                  diagnostics/debug file
1858  C
1859  C    /FFRDAT/ ($include 'FFRDAT.INC')
1860  C     MFIELD = Max. no. of fields that Free-Field-Reader can process
1861  C     NFORM  = Max. length of a CHARACTER data field
1862  C
1863  C  LOCAL/
1864  C     CVALUE = CHARACTER values of the data fields
1865  C     IERR   = INTEGER error flag
1866  C     IOSTAT = INTEGER value for ANSI FORTRAN I/O status
1867  C     IVALUE = INTEGER values of the data fields
1868  C     KVALUE = Translation states of the data fields
1869  C     NFIELD = Number of fields
1870  C     RVALUE = REAL values of the data fields
1871  C
1872  C  EXIT/
1873  C   --through subroutine call
1874  C     NUMVAR = No. of LHS variables
1875  C
1876  C*********************************************************************
1877  C234567
1878        IMPLICIT NONE
1879        INCLUDE 'IOCOM.INC/LIST'
1880        INCLUDE 'FFRDAT.INC/LIST'
1881        INCLUDE 'PARACM.INC/LIST'
1882        INCLUDE 'NEWPAR.INC/LIST'
1883
1884        INTEGER IERR, IOSTAT, NFIELD
```

```
      INTEGER IVALUE(MFIELD), KVALUE(MFIELD)
      REAL RVALUE(MFIELD)
      CHARACTER*(NFORM) CVALUE(MFIELD)

C<><><><><><><><><><><><><>
C...BEGIN PROCEDURES...
C<><><><><><><><><><><><><>

      IERR   = 0

C     ...Begin Scanning PREINFIL control card data batch file
   10 CALL FFRDFLDS( INASCI, NOUTFL, ' ', MFIELD, IOSTAT,
     +              NFIELD, KVALUE, CVALUE, IVALUE, RVALUE )

   20 IF (IOSTAT.LT.0) THEN
C        ...End Of File for PREINFIL control card input file found
         GOTO 100

      ELSEIF (IOSTAT.GT.0) THEN
C        ...Set the error flag, abort after EOF found
         IERR = IERR + 1
C        ...Read next record
         GOTO 10

      ELSEIF (IOSTAT.EQ.0) THEN

         IF ( KVALUE(1).EQ.-1 .OR.
     +       (KVALUE(1).EQ.0 .AND.CVALUE(1).EQ.'!')) THEN
C           ...This is a comment line or a blank w/o information
C           ...Read next record
            GOTO 10

         ELSEIF(KVALUE(1).EQ.0.AND.CVALUE(1)(:5).EQ.'*ECHO')THEN
C           ...This is info. to be echoed directly to INFIL input control
C              file
C*            CALL DOECHO( CVALUE )
C           ...Read next record
            GOTO 10

         ELSEIF(KVALUE(1).EQ.0.AND.CVALUE(1)(:5).EQ.'*FILE')THEN
C           ...allows user to select certain output filenames for INFIL
C              these filenames are inserted into the INFIL input control file
C           ...initialize filenames used by INFIL
               FILESUMM = ' '
               FILEMAIN = ' '
               FILEECHO = ' '
            CALL RDFILNM(IOSTAT, NFIELD, KVALUE, CVALUE, IVALUE, RVALUE)
C           ...Read next record
            GOTO 10

         ELSEIF(KVALUE(1).EQ.0.AND.CVALUE(1)(:8).EQ.'*REPLACE')THEN
C           ...Begin retrieving parameter identity (e.g. sampled by LHS)
C              for replacement into new INFIL input control file
            CALL RDPAR(IOSTAT, NFIELD, KVALUE, CVALUE, IVALUE, RVALUE)
C           ...Read next record
            GOTO 10

```

```
1942            ELSEIF(KVALUE(1).EQ.0.AND.CVALUE(1)(:4).EQ.'*END')THEN
1943   C           ...Found END of PREINFIL control file
1944   C           ...Abort scaning session
1945               GOTO 100
1946
1947            ELSE
1948   C           ...Meaningless data found
1949   C           ...Read next record
1950               GOTO 10
1951            ENDIF
1952         ENDIF
1953
1954    100  IF (IERR.GT.0) THEN
1955            WRITE(NOUTFL,*)' ***',IERR,' ERRORS FOUND IN RDINFL ***'
1956            CALL QAABORT('PROCTL')
1957         ENDIF
1958         RETURN
1959   C-----------------------------------------------------------------------
1960   C**** END OF SUBROUTINE PROCTL ****
1961   C-----------------------------------------------------------------------
1962         END
```

```
1963
1964  *DECK RDFILNM
1965        SUBROUTINE RDFILNM( IOSTAT,NFIELD,KVALUE,CVALUE,IVALUE,RVALUE )
1966  C*********************************************************************
1967  C****                                                             ****
1968  C****            R_ea_D  P_A_R ameter names module                ****
1969  C****            -   -   - - -                                    ****
1970  C*********************************************************************
1971  C
1972  C   PURPOSE:    Reads names of USER SELECTED INFIL output file names
1973  C
1974  C   AUTHOR:     Ron McCurley
1975  C
1976  C   UPDATED:    Aug 13 1999
1977  C
1978  C  CALLED BY:   PRCTRL
1979  C
1980  C  CALLS:       FREFLD
1981  C               QAABORT
1982  C
1983  C  ARGUMENTS:
1984  C  ENTRY/
1985  C    --common blocks
1986  C     /COMMIO/ ($includE 'GI1_COMMIO.INC')
1987  C      ISCRAT = Device no. of PREINFIL scratch file
1988  C      FILEIN = Device no. of PREINFIL input text file
1989  C      NOUTFL = Device no. of diagnostics/debut output file
1990  C
1991  C     /P/ ($includE '.INC')
1992  C
1993  C    --through subroutine call
1994  C      MFIELD = Max. no. of data fields FFR can process
1995  C      IOSTAT = ANSI FORTRAN I/O error flag
1996  C      NFIELD = No. of data fields read by FFR
1997  C      KVALUE = INTEGER array of types of data fields read by FFR.
1998  C      CVALUE = CHARACTER  array of data fields read by FFR.
1999  C      IVALUE = INTEGER    "
2000  C      RVALUE = REAL       "
2001  C
2002  C  LOCAL/
2003  C      none
2004  C
2005  C  EXIT/
2006  C    --common blocks
2007  C     /I/ ($includE 'I.INC') ???
2008  C      ...REAL variables
2009  C
2010  C    --through subroutine call
2011  C      MFIELD = Max. no. of data fields FFR can process
2012  C      IOSTAT = ANSI FORTRAN I/O error flag
2013  C      NFIELD = No. of data fields read by FFR
2014  C      KVALUE = INTEGER array of types of data fields read by FFR.
2015  C      CVALUE = CHARACTER  array of data fields read by FFR.
2016  C      IVALUE = INTEGER    "
2017  C      RVALUE = REAL       "
2018  C
2019  C*********************************************************************
```

```
2020  C234567
2021        IMPLICIT NONE
2022        INCLUDE 'FFRDAT.INC/LIST'
2023        INCLUDE 'IOCOM.INC/LIST'
2024        INCLUDE 'PARACM.INC/LIST'
2025        INCLUDE 'NEWPAR.INC/LIST'
2026
2027        INTEGER I, IERR, IOSTAT, K, NFIELD
2028        INTEGER IVALUE(*), KVALUE(*)
2029        REAL RVALUE(*)
2030        CHARACTER*(NFORM) CVALUE(*)
2031        LOGICAL NAMED_REG
2032
2033  C<><><><><><><><><><><><>
2034  C...BEGIN PROCEDURES...
2035  C<><><><><><><><><><><><>
2036
2037        IERR   = 0
2038
2039  C     ...Read data for parameter replacement into new INFIL input control
2040  C        file
2041    10  CALL FFRDFLDS( INASCI, NOUTFL, ' ', MFIELD, IOSTAT,
2042       +                NFIELD, KVALUE, CVALUE, IVALUE, RVALUE )
2043
2044    20  IF (IOSTAT.LT.0) THEN
2045  C        ...End Of File for PREINFIL control card input file found
2046           WRITE(NOUTFL,1001)
2047           WRITE(NOUTFL,1002)
2048           WRITE(NOUTFL,1001)
2049           GO TO 100
2050
2051        ELSEIF (IOSTAT.GT.0) THEN
2052  C        ...Set the error flag, abort after EOF found
2053           IERR = IERR + 1
2054  C        ...Read next record
2055           GOTO 10
2056
2057        ELSEIF (IOSTAT.EQ.0) THEN
2058
2059           IF ( KVALUE(1).EQ.-1 .OR.
2060       +      (KVALUE(1).EQ.0 .AND.CVALUE(1).EQ.'!')) THEN
2061  C          ...This is a comment line or a blank w/o information
2062  C          ...Read next record
2063             GOTO 10
2064
2065           ELSEIF( KVALUE(1).EQ.0.AND.CVALUE(1)(:4).EQ.'NAME' .OR.
2066       +           CVALUE(1)(:4).EQ.'name' ) THEN
2067             DO 30, I=2,NFIELD-1,2
2068                IF( KVALUE(I).EQ.0.AND.CVALUE(I)(:4).EQ.'SUMM' .OR.
2069       +            CVALUE(I)(1:4).EQ.'summ') THEN
2070  C                ...get name of main output file containing net infiltration
2071  data
2072                  FILESUMM = CVALUE(I+1)
2073
2074                ELSEIF( KVALUE(I).EQ.0.AND.CVALUE(I)(:4).EQ.'MAIN' .OR.
2075       +            CVALUE(I)(1:4).EQ.'main') THEN
```

```
2076  C                 ...get name of main output file containing net infiltration
2077  data
2078                    FILEMAIN = CVALUE(I+1)
2079
2080                 ELSEIF(KVALUE(I).EQ.0.AND.CVALUE(I)(:4).EQ.'ECHO' .OR.
2081       +               CVALUE(I)(:4).EQ.'echo') THEN
2082  C                ...get name of output file echoing input to INFIL
2083                   FILEECHO = CVALUE(I+1)
2084                 ENDIF
2085     30        CONTINUE
2086  C            ...Read next keyword
2087               GOTO 100
2088             ELSE
2089  C          ...Found meaningless data
2090               WRITE(NOUTFL,*)' ***found unexpected meaningless data ',
2091       +                       'may be a problem in user input file!***'
2092  C            ...Read next record
2093               GOTO 10
2094             ENDIF
2095  C          ...Read next record
2096             GOTO 10
2097         ENDIF
2098  C
2099    100 CONTINUE
2100  C
2101         RETURN
2102   1001 FORMAT('*****************************************')
2103   1002 FORMAT('WARNING! '/
2104       +       'Encountered unexpected end of user input'/
2105       +       'May be bad file! ')
2106  C-----------------------------------------------------------------------
2107  C**** END OF SUBROUTINE RDFILNM  ****
2108  C-----------------------------------------------------------------------
2109         END
```

```
2110
2111  *DECK RDINFTMP
2112        SUBROUTINE RDINFTMP(NUMREC, NUMFLD, INFDAT, RINFDAT, CINFDAT)
2113  C***********************************************************************
2114  C
2115  C  ReaDs INFIL TeMPlate file and stores lines/data
2116  C
2117  C***********************************************************************
2118  C
2119        INCLUDE 'PARACM.INC/LIST'
2120        INCLUDE 'FFRDAT.INC/LIST'
2121
2122        INTEGER IERR, IOSTAT, NFIELD, NUMREC
2123        INTEGER INFDAT(MAXREC,*), IVALUE(MFIELD), KVALUE(MFIELD),
2124       +        NUMFLD(MAXREC)
2125        REAL RVALUE(MFIELD)
2126        REAL RINFDAT(MAXREC,*)
2127        CHARACTER*(NFORM) CVALUE(MFIELD)
2128        CHARACTER*80 CINFDAT(MAXREC,*)
2129
2130        INCLUDE 'NEWPAR.INC/LIST'
2131        INCLUDE 'IOCOM.INC/LIST'
2132
2133  C ***Begin reading template for Input Control File (ICF) for INFIL
2134        ILINE = 0
2135
2136     10 CONTINUE
2137          ILINE = ILINE + 1
2138  C     ...Begin Scanning PREINFIL template INFIL control file
2139    15    CALL FFRDFLDS( INPTMP, NOUTFL, ' ',    MFIELD, IOSTAT,
2140       +                 NFIELD, KVALUE, CVALUE, IVALUE, RVALUE )
2141
2142    20    IF (IOSTAT.LT.0) THEN
2143  C          ...End Of File for PREINFIL template INFIL control file found
2144             GOTO 100
2145
2146          ELSEIF (IOSTAT.GT.0) THEN
2147  C          ...Set the error flag, abort after EOF found
2148             IERR = IERR + 1
2149  C          ...Read next record
2150             GOTO 10
2151
2152          ELSEIF (IOSTAT.EQ.0) THEN
2153             NUMFLD(ILINE) = NFIELD
2154             DO 30 I=1,NFIELD
2155               IF ( KVALUE(I).EQ.-1 .OR. KVALUE(I).EQ.0 ) THEN
2156  C            ...This is a comment field or a blank w/o information
2157                 CINFDAT(ILINE,I) = CVALUE(I)
2158               ELSEIF( KVALUE(I).EQ.1 ) THEN
2159  C            ...This is real data
2160                 RINFDAT(ILINE,I) = RVALUE(I)
2161               ELSEIF( KVALUE(I).EQ.2 ) THEN
2162  C            ...This is integer data
2163                 INFDAT(ILINE,I) = IVALUE(I)
2164
2165               ENDIF
2166    30       CONTINUE
```

```
2167          ENDIF
2168
2169          IF (ILINE.GT.MAXREC) THEN
2170             PRINT *, 'The number of lines is greater than ',
2171       +                      'allowed (',MAXREC,')'
2172             CALL QAABORT('(RDINFTMP) - reading INFIL ICF ***')
2173          ENDIF
2174  C        ...Read next record
2175          GO TO 10
2176    100 CONTINUE
2177
2178        NUMREC = ILINE
2179  C
2180        RETURN
2181  C
2182        END
```

```
2183
2184  *DECK RDPAR
2185        SUBROUTINE RDPAR( IOSTAT, NFIELD, KVALUE, CVALUE, IVALUE, RVALUE )
2186  C***********************************************************************
2187  C****                                                               ****
2188  C****            R_ea_D  P_A_R ameter names module                  ****
2189  C****            -    -  - - -                                      ****
2190  C***********************************************************************
2191  C
2192  C   PURPOSE:    Reads names of new INFIL input CONTROL parameters
2193  C               as stored (with values) in CDB files as matched with
2194  C               key user names (fixed or sampled)
2195  C
2196  C   AUTHOR:     Ron McCurley
2197  C
2198  C   UPDATED:    May 1999
2199  C
2200  C  CALLED BY:   PRCTRL
2201  C
2202  C  CALLS:       FREFLD
2203  C               QAABORT
2204  C
2205  C  ARGUMENTS:
2206  C  ENTRY/
2207  C    --common blocks
2208  C     /COMMIO/ ($includE 'GI1_COMMIO.INC')
2209  C      ISCRAT = Device no. of PREINFIL scratch file
2210  C      FILEIN = Device no. of PREINFIL input text file
2211  C      NOUTFL = Device no. of diagnostics/debut output file
2212  C
2213  C     /PGENII/ ($includE 'GI1_PGENII.INC')
2214  C
2215  C    --through subroutine call
2216  C      MFIELD = Max. no. of data fields FFR can process
2217  C      IOSTAT = ANSI FORTRAN I/O error flag
2218  C      NFIELD = No. of data fields read by FFR
2219  C      KVALUE = INTEGER array of types of data fields read by FFR.
2220  C      CVALUE = CHARACTER  array of data fields read by FFR.
2221  C      IVALUE = INTEGER    "
2222  C      RVALUE = REAL       "
2223  C
2224  C  LOCAL/
2225  C      none
2226  C
2227  C  EXIT/
2228  C    --common blocks
2229  C     /I/ ($includE 'I.INC') ???
2230  C      ...REAL variables
2231  C
2232  C    --through subroutine call
2233  C      MFIELD = Max. no. of data fields FFR can process
2234  C      IOSTAT = ANSI FORTRAN I/O error flag
2235  C      NFIELD = No. of data fields read by FFR
2236  C      KVALUE = INTEGER array of types of data fields read by FFR.
2237  C      CVALUE = CHARACTER  array of data fields read by FFR.
2238  C      IVALUE = INTEGER    "
2239  C      RVALUE = REAL       "
```

```
2240  C
2241  C*********************************************************************
2242  C234567
2243        IMPLICIT NONE
2244        INCLUDE 'FFRDAT.INC/LIST'
2245        INCLUDE 'IOCOM.INC/LIST'
2246        INCLUDE 'PARACM.INC/LIST'
2247        INCLUDE 'NEWPAR.INC/LIST'
2248        INCLUDE 'INFIL.INC/LIST'
2249
2250        INTEGER I, IERR, IOSTAT, K, NFIELD
2251        INTEGER IVALUE(*), KVALUE(*)
2252        REAL RVALUE(*)
2253        CHARACTER*(NFORM) CVALUE(*)
2254        LOGICAL NAMED_REG
2255
2256  C<><><><><><><><><><><><>
2257  C...BEGIN PROCEDURES...
2258  C<><><><><><><><><><><><>
2259
2260        IERR   = 0
2261
2262  C     ...Read data for parameter replacement into new INFIL input control
2263  C        file
2264    10  CALL FFRDFLDS( INASCI, NOUTFL, ' ', MFIELD, IOSTAT,
2265       +               NFIELD, KVALUE, CVALUE, IVALUE, RVALUE )
2266
2267    20  IF (IOSTAT.LT.0) THEN
2268  C        ...End Of File for PREINFIL control card input file found
2269           WRITE(NOUTFL,1001)
2270           WRITE(NOUTFL,1002)
2271           WRITE(NOUTFL,1001)
2272           GO TO 100
2273
2274        ELSEIF (IOSTAT.GT.0) THEN
2275  C        ...Set the error flag, abort after EOF found
2276           IERR = IERR + 1
2277  C        ...Read next record
2278           GOTO 10
2279
2280        ELSEIF (IOSTAT.EQ.0) THEN
2281
2282           IF ( KVALUE(1).EQ.-1 .OR.
2283       +      (KVALUE(1).EQ.0 .AND.CVALUE(1).EQ.'!')) THEN
2284  C          ...This is a comment line or a blank w/o information
2285  C          ...Read next record
2286             GOTO 10
2287
2288           ELSEIF(KVALUE(1).EQ.0.AND.CVALUE(1)(:6).EQ.'REGION' .OR.
2289       +          CVALUE(1)(1:6).EQ.'region') THEN
2290  C          ...get region name in CDB's where infitration parameters are
2291             REGION=CVALUE(2)
2292             NAMED_REG =.TRUE.
2293  C          ...Read next record
2294             GOTO 10
2295
2296           ELSEIF(KVALUE(1).EQ.0.AND.CVALUE(1)(:4).EQ.'PROP' .OR.
```

```
2297        +          CVALUE(1)(:4).EQ.'prop') THEN
2298  C            ...Begin retrieving parameter identity (e.g. sampled by LHS)
2299  C               for replacement into new INFIL input control file;
2300  C               loop over remaining words in this field
2301              DO 40 I=2,NFIELD,2
2302  C               ...get CDB names of variables to retrieve their LHS sampled
2303  C                  values
2304                  K = K+1
2305                  IF(K.GT.MAXVAR) THEN
2306                    WRITE(NOUTFL,*)'*** Exceeded max number of NEW (LHS ',
2307        +                          'or fixed parameters at KEYWORD = ',
2308        +                          CVALUE(I),' Check PARACM.INC!'
2309                    CALL QAABORT('Exceeded max number of new (LHS or ',
2310        +                        'fixed parameters, check PARACM.INC' )
2311                  ENDIF
2312
2313  C               ...INFIL_NAME is the name of the parameter as designated in
2314  C                  INFIL.INC; CDB_NAME is the name of the same parameter as
2315  C                  designated in the suite of
2316  C                  CDB files produced by POSTLHS; these may be identical
2317                  INFIL_NAME(K) = CVALUE(I)
2318                  CDB_NAME(K) = CVALUE(I+1)
2319                  IF(CDB_NAME(K)(1:4).EQ.'SAME' .OR.
2320        +            CDB_NAME(K)(1:4).EQ.'same') CDB_NAME(K)=INFIL_NAME(K)
2321
2322    40        CONTINUE
2323  C           ...Read next record
2324              GOTO 10
2325            ELSE
2326  C         ...Found meaningless data
2327              WRITE(NOUTFL,*)' ***found unexpected meaningless data ',
2328        +                    'may be a problem in user input file!***'
2329  C           ...Read next record
2330              GOTO 10
2331            ENDIF
2332  C         ...Read next record
2333            GOTO 10
2334          ENDIF
2335
2336   100    NUMVAR = K
2337          IF(.NOT. NAMED_REG) REGION = 'REGION00'
2338  C
2339          RETURN
2340   1001   FORMAT('*****************************************')
2341   1002   FORMAT('WARNING! '/
2342        +        'Encountered unexpected end of user input'/
2343        +        'May be bad file! ')
2344  C-----------------------------------------------------------------------
2345  C**** END OF SUBROUTINE RDPAR  ****
2346  C-----------------------------------------------------------------------
2347          END
```

```
2348
2349  *DECK RECHEA
2350        SUBROUTINE RECHEA( Q,       C,       IDEBLK, NUMPRP, IASPRP,
2351       $                   XMATPR, QAINFO, NAMELB, NMATPR )
2352  C*********************************************************************
2353  C****                                                             ****
2354  C****          R_E_ad  C_amdat  H_E_A_der  section  routine       ****
2355  C****          - -     -        - - -                             ****
2356  C*********************************************************************
2357  C
2358  C  PURPOSE:    Reads pertinent "HEADER" sections of the input
2359  C              CAMDAT file into common blocks and arrays
2360  C
2361  C  AUTHOR:     Jonathan S. Rath
2362  C
2363  C  UPDATED:    13-JUN-1994  --JSR: First Edition
2364  C
2365  C  CALLED BY:  PROCDB
2366  C
2367  C  CALLS:      DBIELBLK
2368  C              DBIPROP
2369  C              DBIPRNM
2370  C              DBIQAREC
2371  C              DBINSTEP
2372  C              QAABORT
2373  C
2374  C  ARGUMENTS:
2375  C  ENTRY/
2376  C   --common blocks
2377  C    /IOCOM/ (from FORTRAN statement: INCLUDE 'IOCOM.INC')
2378  C     ICDB    = Device no of Input CAMDAT File
2379  C
2380  C    /GEOPAR/ (from FORTRAN statement: INCLUDE 'GEOPAR.INC')
2381  C     NELBLK = Number of ELement BLocKs in mesh (i.e. # of materials)
2382  C     NQAREC = Number of QA records
2383  C     NSTEP  = No. of time steps in input CAMDAT (INTEGER)
2384  C     NVAREL = No. of element variables (INTEGER)
2385  C     NUQPRP = No. of UniQUe PRoperty names (INTEGER)
2386  C
2387  C   --through subroutine call
2388  C     Q      = REAL type base dynamic array
2389  C     C      = CHARACTER*1 type base dynamic array
2390  C
2391  C  LOCAL/
2392  C     IERR   = INTEGER error code
2393  C
2394  C  EXIT/
2395  C   --through subroutine call
2396  C     IDEBLK = INTEGER ID array of matrial regions (INTEGER)
2397  C     NUMPRP = No. of properties per material region (INTEGER)
2398  C     IASPRP = Logical array for properties in materials
2399  C     XMATPR = Properties array (REAL)
2400  C     QAINFO = Quality Assurance array (CHAR*8)
2401  C     NAMELB = Names of the material regions (CHAR*8)
2402  C     NMATPR = Names of the properties (CHAR*8)
2403  C
2404  C*********************************************************************
```

```
2405  C234567
2406        IMPLICIT NONE
2407        INCLUDE 'IOCOM.INC'
2408        INCLUDE 'GEOPAR.INC'
2409        INTEGER  I, IERR, J, K, NHISTP
2410        INTEGER IDEBLK(*), NUMPRP(*)
2411        REAL Q(*), XMATPR(*)
2412        CHARACTER*(1) C(*)
2413        CHARACTER*(8) NAMELB(*), NMATPR(*), QAINFO(*)
2414        LOGICAL IASPRP(*)
2415  C
2416  C<><><><><><><><><><><><>
2417  C...Begin Procedures...
2418  C<><><><><><><><><><><><>
2419  C
2420        IERR = 0
2421
2422  C     ...Read material ID and names information
2423        CALL DBIELBLK( ICDB, NELBLK, NAMELB, IDEBLK, IERR )
2424        IF(IERR.NE.0)
2425       $CALL QAABORT('(RECHEA) - reading material ID and names array ***')
2426
2427  C     ...Read property names
2428        CALL DBIPRNAM( ICDB, NUQPRP, NMATPR, Q, C, IERR )
2429        IF(IERR.NE.0)
2430       $CALL QAABORT('(RECHEA) - reading PROPERTY NAMES array ***')
2431
2432  C     ...Read property data
2433        DO 10 I=1,NUQPRP
2434           J = (I-1)*NELBLK+1
2435           CALL DBIPROP( ICDB, '*', I, IASPRP(J), XMATPR(J),
2436       $                 Q,         C, IERR )
2437           IF(IERR.NE.0)
2438       $   CALL QAABORT('(RECHEA) - reading PROPERTY values ***')
2439    10  CONTINUE
2440
2441  C     ...Construct the number of properties per material array
2442        DO 30 J=1,NELBLK
2443           NUMPRP(J) = 0
2444           DO 20 I=1,NUQPRP
2445              K = (I-1)*J+1
2446              IF(IASPRP(K))NUMPRP(J) = NUMPRP(J)+1
2447    20     CONTINUE
2448    30  CONTINUE
2449
2450  C     ...Read QA data
2451        CALL DBIQAREC( ICDB, NQAREC, QAINFO, IERR )
2452        IF(IERR.NE.0)CALL QAABORT('(RECHEA) - reading QA array ***')
2453
2454  C     ...Get the number of time steps (NSTEP)
2455        CALL DBINSTEP( ICDB, NSTEP, NHISTP, IERR )
2456        IF(IERR.NE.0)CALL QAABORT('(RECHEA) - obtaining input CAMDAT '//
2457       $                          'no. of time steps ***')
2458
2459        RETURN
2460  C-----------------------------------------------------------------------
2461  C**** END OF SUBROUTINE RECHEA *****
```

```
C-----------------------------------------------------------------------
      END
```

```
*DECK TRNDAT
      SUBROUTINE TRNDAT( NUMPRP, XMATPR, NAMELB, NMATPR, XINFILVAL )
C     ***Modified 8 Nov 99 to account for sampling for lognormal distribution
C        for etcoeffa with negative values RDM
C*********************************************************************
C

      INCLUDE 'GEOPAR.INC/LIST'
      INCLUDE 'PARACM.INC/LIST'
      INCLUDE 'INFIL.INC/LIST'
      INCLUDE 'NEWPAR.INC/LIST'

      INTEGER  I, IBLK, IERR, INDEXJ, J, K
      INTEGER  ISUM(NELBLK), NUMPRP(*)
      REAL XINFILVAL(*), XMATPR(*)
      CHARACTER*(8) NAMELB(*), NMATPR(*)
      LOGICAL MATCH(NUMVAR)

C***  translate material property values extracted from the input
C     CDB file to appropriate parameter values for the INFIL input
C     control file

      DO 100 IBLK=1,NELBLK

        IF(REGION .EQ. NAMELB(IBLK)) THEN

          DO 80 J=1,NUMPRP(IBLK)

            INDEXJ = IBLK + NELBLK*(J-1)

            DO 50 I=1,NUMVAR
              MATCH(I) = .FALSE.

              IF(NMATPR(J) .EQ. CDB_NAME(I)) THEN
                XINFILVAL(I)=XMATPR(INDEXJ)
C               ...change sign of etcoeffa to negative
                IF(CDB_NAME(I)(1:8).EQ.'ETCOEFFA') THEN
                   XINFILVAL(I) = -1.0*(XINFILVAL(I))
                   MATCH(I)=.TRUE.
C               ...change value of 'small' negative sampled variables to
C                  zero where LHS picked below the 99 percentile
                ELSEIF( CDB_NAME(I)(1:7).EQ.'BRPOROS' .AND.
     +                  XINFILVAL(I).LE.0.0 ) THEN
                   XINFILVAL(I) = 1.0E-3
                ELSEIF( CDB_NAME(I)(1:8).EQ.'BRZDEPTH' .AND.
     +                  XINFILVAL(I).LE.0.0 ) THEN
                   XINFILVAL(I) = 1.0E-1
                ENDIF
                GOTO 80
              ENDIF
   50       CONTINUE
   80     CONTINUE
        ENDIF

 100  CONTINUE
```

```
2521        RETURN
2522        END
```

```
2523  *DECK,TYPRQS
2524        SUBROUTINE TYPRQS( PROMPT, ISTRNG )
2525  C*********************************************************************
2526  C
2527  C  PURPOSE:      Outputs a request for a character string using PROMPT
2528  C                and then inputs the character string (ISTRNG) in an
2529  C                interactive session.
2530  C
2531  C  AUTHOR:       Rob Rechard
2532  C
2533  C  UPDATED:      June 1985
2534  C                July 1987          --Ginger Wilkinson
2535  C                15 February, 1989  --Jonathan S. Rath made more generic
2536  C
2537  C  CALLED BY:    PREPRO
2538  C
2539  C  ARGUMENTS:
2540  C  ENTRY/
2541  C   --through subroutine call
2542  C     PROMPT  = Message to print on the screen
2543  C
2544  C  EXIT/
2545  C   --subroutine call
2546  C     ISTRNG  = Character string read
2547  C
2548  C*********************************************************************
2549  C234567
2550        IMPLICIT NONE
2551        CHARACTER*(*) ISTRNG, PROMPT
2552        WRITE(*,1000)PROMPT
2553     10 READ(*,'(A)',END=20,ERR=30) ISTRNG
2554        RETURN
2555
2556     20 WRITE(*,2000)
2557        GOTO 10
2558     30 WRITE(*,3000)
2559        GOTO 10
2560
2561  C---------------- F O R M A T   S T A T E M E N T S -------------------
2562  1000  FORMAT(A,'  >>')
2563  2000  FORMAT(' ***NO DATA--TRY AGAIN***')
2564  3000  FORMAT(' ***BAD CHARACTER STRING--TRY AGAIN***')
2565  C---------------------------------------------------------------------
2566  C**** END OF SUBROUTINE TYPRQS ****
2567  C---------------------------------------------------------------------
2568        END
```

```
2569
2570  *DECK,WELCOM
2571        SUBROUTINE WELCOM
2572  C*********************************************************************
2573  C
2574  C  PURPOSE:      Produces Instructions for main program usage
2575  C
2576  C  PROGRAMMER:   Jonathan S. Rath
2577  C
2578  C  UPDATED:      24 May, 1989 --First Ed.
2579  C
2580  C  CALLED BY:    PREPRO
2581  C
2582  C  CALLS:        QABANNER
2583  C
2584  C  ARGUMENTS:
2585  C  ENTRY/
2586  C    --common blocks
2587  C
2588  C    /QACOMMON/
2589  C     PROGRM  =  The program name (CHAR*12)
2590  C     PVERSN  =  The program version number (CHAR*8)
2591  C
2592  C    /L1/  Contains Line 1 of a 3-line program discription written out
2593  C          following the program banner
2594  C    /L2/  Contains Line 2 of a 3-line program discription written out
2595  C          following the program banner
2596  C    /L3/  Contains Line 3 of a 3-line program discription written out
2597  C          following the program banner
2598
2599  C  LOCAL/
2600  C     IOUT   =  Device number of output file
2601  C
2602  C  EXIT/
2603  C      none
2604  C
2605  C*********************************************************************
2606  C234567
2607
2608        IMPLICIT NONE
2609
2610        INCLUDE 'CAMCON_COMMON.INC/list'
2611
2612        INTEGER IOUT
2613        CHARACTER*1  CHAR
2614        CHARACTER*80 LINE1,LINE2,LINE3
2615        COMMON /L1/  LINE1
2616        COMMON /L2/  LINE2
2617        COMMON /L3/  LINE3
2618        DATA IOUT/5/
2619
2620  C<><><><><><><><><><><><>
2621  C...Begin Procedures...
2622  C<><><><><><><><><><><><>
2623
2624        OPEN(IOUT,FILE='SYS$OUTPUT',FORM='FORMATTED',STATUS='UNKNOWN')
2625        CALL QABANNER(IOUT,LINE1,LINE2,LINE3)
```

```
2626          WRITE(IOUT,1000)
2627    C     ...Abort PROGRM program execution ?
2628          WRITE(IOUT,1200)PROGRM,PVERSN
2629          READ(IOUT,'(A)')CHAR
2630          IF(CHAR.NE.' ')THEN
2631             STOP '**** USER ABORTED EXECUTION IN SUBROUTINE WELCOM ****'
2632          ENDIF
2633          CLOSE(IOUT)
2634          RETURN
2635    C
2636    C--------------- F O R M A T   S T A T E M E N T S -------------------
2637    C
2638    1000  FORMAT(///
2639         +'*************************************************************'/
2640         +'    PREINFIL: PRE-processor for INFIL input control file    '/
2641         +'*************************************************************'///
2642         +' Following are prompts for'//
2643         +'      (1) Filename of PREINFIL control input file'/
2644         +'      (2) Filename of Computational Data Base to be read'/
2645         +'      (3) Filename of Template INFIL input control file '/
2646         +'      (3) Filename of PREINFIL generated INFIL input file'/
2647         +'      (4) (OPTIONAL) Filename of PREINFIL diagnostics/debug'/
2648         +'          file'//,
2649         +'*************************************************************'/)
2650
2651    1200  FORMAT(/79('*'),///1X,'To CONTINUE program ',A,' V',A,' press',
2652         +       ' the RETURN key.',/1X,'To ABORT program, type the word',
2653         +       ' ABORT')
2654    C
2655    C-----------------------------------------------------------------------
2656    C**** END OF SUBROUTINE WELCOM ****
2657    C-----------------------------------------------------------------------
2658          END
```

```
2659
2660  *DECK WRIINF
2661        SUBROUTINE WRIINF(FCDB,NUMREC,NUMFLD,INFDAT,RINFDAT,CINFDAT,
2662     +                    INFIL_VAL)
2663  C*********************************************************************
2664  C***                                                               ***
2665  C***       WRIte INFIL input control file                          ***
2666  C***                                                               ***
2667  C*********************************************************************
2668  C
2669  C   PURPOSE:  Writes new INFIL input control file using both unmodified
2670  C             data from a template input control file and new values
2671  C             from a designated computational data base and specified
2672  C             parameters
2673  C
2674  C   AUTHOR:   Ron McCurley
2675  C
2676  C   UPDATED:  25 May, 1999 --Original version
2677  C              5 Aug, 1999 --fixed format bug in record 11 and in record 15
2678  C             10 Nov, 1999 --modified for ndaymap files
2679  C
2680  C   CALLED BY:   PREINF (MAIN program)
2681  C
2682  C   CALLS:       QAABORT
2683  C
2684  C   LIMITATION:  This routine writes input specifically for a
2685  C                 particular version of the INF code.
2686  C
2687  C   ARGUMENTS:
2688  C   ENTRY/
2689  C     --common block
2690  C      /IOCOM.INC/ ($include 'IOCOM.INC')
2691  C       NOUTFL = File unit of diagnostics/debug file
2692  C       INPINF = File unit of the INF input file
2693  C
2694  C      /CAMCON_COMMON.INC/ ($include 'CAMCON_COMMON.INC')
2695  C       RUNDAT, RUNTIM - the run date and time
2696  C
2697  C     --through subroutine call
2698  C       NUMVAR = No. of INF varaibles
2699  C       INFVAR = INF material and parameter ID array
2700  C       FCDB   = Name of Computational database file.
2701  C
2702  C   LOCAL/
2703  C       IERR   = INTEGER error flag
2704  C
2705  C  EXIT/
2706  C     --subroutine call
2707  C       INFDIS = INF variable distribution name array
2708  C       INFVRN = Lower & Upper distribution ranges of INF varaible array
2709  C
2710  C*********************************************************************
2711  C234567
2712        IMPLICIT NONE
2713
2714        INCLUDE 'CAMCON_COMMON.INC/list'
2715        INCLUDE 'FFRDAT.INC/LIST'
```

```
2716          INCLUDE 'PARACM.INC/LIST'
2717          INCLUDE 'IOCOM.INC/LIST'
2718          INCLUDE 'NEWPAR.INC/LIST'
2719          INCLUDE 'INFIL.INC/LIST'
2720
2721          INTEGER  I,   I1,  I2,  I3,  ICHAR,  J,   K,   IERR,  IFLD,
2722         +         LASTREC, NDAYMAP, NFLD,  NUMREC
2723          INTEGER  INFDAT(MAXREC,*), NUMFLD(*)
2724
2725          REAL     INFIL_VAL(*), RINFDAT(MAXREC,*)
2726          CHARACTER*80 CINFDAT(MAXREC,*)
2727          CHARACTER*80 FCDB
2728
2729          LOGICAL  ERROR, GOTALL, NEWREC(100)
2730
2731    C<><><><><><><><><><><><>
2732    C...BEGIN PROCEDURES...
2733    C<><><><><><><><><><><><>
2734
2735          IERR   = 0
2736          WRITE(NOUTFL,900)
2737
2738    C...Assemble new arrays INFDAT, RINFDAT
2739    C   (only line 3 thru 18 may contain new paramemter data)--this may change!
2740    C   so may want to make this another user input parameter for PREINFIL
2741          I1 = 3
2742          I2 = 26
2743          LASTREC = 0
2744          NDAYMAP = INFDAT(27,1)
2745          I3 = I2 + 2*NDAYMAP + 8
2746    C
2747          DO 200 J = I1,I2
2748             NEWREC(J) = .FALSE.
2749    C        ...loop over fields in record, starting with 3 (no comments
2750    C           until at least the 3rd field)
2751             DO 190 IFLD=3,NUMFLD(J)
2752               ICHAR = 8
2753               IF ( CINFDAT(J,IFLD)(8:8) .EQ. ')' ) ICHAR = 7
2754    C          ...loop over parameters changed
2755               DO 100 K = 1, NUMVAR
2756    C*              IF(J.EQ.18 .AND. INFIL_NAME(K)(1:5).EQ.'ATEMP1')
2757    C*      +          RINFDAT(J,2) = INFIL_VAL(K)
2758    C            ...first ICHAR chars is enough
2759                 IF( INFIL_NAME(K)(1:ICHAR) .EQ. CINFDAT(J,IFLD)(1:ICHAR) )
2760         +         THEN
2761    C                ...this is the name of a modified parameter
2762                     NFLD = IFLD-3
2763                     IF( CINFDAT(J,5)(1:5) .EQ. 'MULTI' .OR.
2764         +               CINFDAT(J,5)(1:5) .EQ. 'multi') NFLD=IFLD-6
2765                     IF( CINFDAT(J,5)(1:4) .EQ. 'BARE' .OR.
2766         +               CINFDAT(J,5)(1:4) .EQ. 'bare') NFLD=IFLD-2
2767                     IF( J .EQ.14) NFLD = 3
2768    C                ...modified integer, real data
2769                     RINFDAT(J,NFLD) = INFIL_VAL(K)
2770    C                ...substitution has been made in this field
2771                     NEWREC(J) = .TRUE.
2772                     GO TO 190
```

```
2773                ENDIF
2774
2775     100      CONTINUE
2776     190    CONTINUE
2777     200 CONTINUE
2778
2779  C**** Write out the new INFIL input control file
2780        DO 300 I=1,I3+1
2781           IF(I.LE.I3) THEN
2782              IF(I .EQ. (28+2*NDAYMAP) .AND. FILEMAIN(:1).NE.' ') THEN
2783                 NEWREC(I) = .TRUE.
2784                 CALL GTVECT(FCDB,FILESUMM)
2785                 WRITE(INPINF,'(A)') FILESUMM
2786              ELSEIF(I .EQ. (29+2*NDAYMAP) .AND.
2787        +             FILEMAIN(:1).NE.' ') THEN
2788                 NEWREC(I) = .TRUE.
2789                 CALL GTVECT(FCDB,FILEMAIN)
2790                 WRITE(INPINF,'(A)') FILEMAIN
2791              ELSEIF(I .EQ. (33+2*NDAYMAP) .AND.
2792        +             FILEECHO(:1).NE.' ') THEN
2793                 NEWREC(I) = .TRUE.
2794                 CALL GTVECT(FCDB,FILEECHO)
2795                 WRITE(INPINF,'(A)') FILEECHO
2796              ENDIF
2797  C
2798              IF(NEWREC(I) .EQ. .TRUE.) THEN
2799  C           ..write out new record
2800                IF(I.EQ.3)
2801        +         WRITE(INPINF,1001) (INFDAT(I,IFLD),IFLD=1,2),
2802        +                            RINFDAT(I,3),
2803        +                            (CINFDAT(I,IFLD),IFLD=4,6)
2804                IF(I.EQ.4)
2805        +         WRITE(INPINF,1002) INFDAT(I,1),
2806        +                            (RINFDAT(I,IFLD),IFLD=2,3),
2807        +                            (CINFDAT(I,IFLD),IFLD=4,6)
2808                IF(I.EQ.10)
2809        +         WRITE(INPINF,1003) (RINFDAT(I,IFLD),IFLD=1,4),
2810        +                            (CINFDAT(I,IFLD),IFLD=5,10)
2811                IF(I.EQ.11)
2812        +         WRITE(INPINF,1004) RINFDAT(I,1),INFDAT(I,2),
2813        +                            RINFDAT(I,3),
2814        +                            (CINFDAT(I,IFLD),IFLD=4,6)
2815                IF(I.EQ.14)
2816        +         WRITE(INPINF,1007) (RINFDAT(I,IFLD),IFLD=1,4),
2817        +                            (CINFDAT(I,IFLD),IFLD=5,8)
2818                IF(I.EQ.15)
2819        +         WRITE(INPINF,1005) (RINFDAT(I,IFLD),IFLD=1,3),
2820        +                            (CINFDAT(I,IFLD),IFLD=4,6)
2821                IF(I.EQ.17)
2822        +         WRITE(INPINF,1006) (RINFDAT(I,IFLD),IFLD=1,2),
2823        +                            (CINFDAT(I,IFLD),IFLD=3,4)
2824                IF(I.EQ.18)
2825        +         WRITE(INPINF,1006) (RINFDAT(I,IFLD),IFLD=1,3),
2826        +                            (CINFDAT(I,IFLD),IFLD=4,10)
2827
2828              ELSE
2829  C             ...echo the old record from the template
```

```
2830                 CALL ECHOREC(INPTMP, INPINF, I)
2831              ENDIF
2832           ELSE
2833  C        ...just echo the rest of the file
2834              CALL DOECHO(INPTMP, INPINF, I, I2+1)
2835           ENDIF
2836   300  CONTINUE
2837
2838  C*          IERR=IERR+1
2839  C*          WRITE (NOUTFL,930)
2840  C*          GO TO 100
2841
2842  C**** Mention errors ****
2843
2844        IF (IERR .GT. 0) THEN
2845           WRITE(NOUTFL,*) ' **** ', IERR, ' Errors found in WRIINF ****'
2846           CALL QAABORT('WRIINF')
2847        ENDIF
2848
2849        RETURN
2850
2851  C---------------- F O R M A T   S T A T E M E N T S ------------------
2852  C
2853   900  FORMAT( //,1X,79('*'),/,1X,'(WRIINF)',// )
2854   930  FORMAT(' *** Material=',A,'; Property/Parameter=',A,'; not found',
2855       +        ' in CDB ***')
2856   1001 FORMAT(2I4,1X,1PE8.2,31X,3A8)
2857   1002 FORMAT(I3,1X,2(1PE8.2,1X),26X,3A8)
2858   1003 FORMAT(4(1PE8.2,1X),12X,6A8)
2859   1004 FORMAT(1PE8.2,I4,2X,1PE8.2,26X,4A8)
2860   1005 FORMAT(3(1PE8.2,2X),22X,3(A8,1X))
2861   1006 FORMAT(2F8.3,32X,2(A8,1X))
2862   1007 FORMAT(4(1X,F5.2),26X,4A8)
2863  C
2864  C**** END OF SUBROUTINE WRIINF ****
2865  C
2866        END
2867  C-----------------------------------------------------------------------
```

```
2868
2869
2870  *COMDECK ARRAY
2871  C*********************************************************************
2872  C     Q-array   include file:
2873  C     UPDATED:  10 OCTOBER, 1988   --Jonathan S. Rath
2874  C     Q         Dynamic Allocation array (REAL)
2875  C     IQ        "                        (INTEGER)
2876  C     CQ        "                        (CHAR*4)
2877  C     MAXQ      Total number of storage words allocated
2878  C*********************************************************************
2879  C234567
2880        INTEGER     MAXC, MAXQ
2881        REAL        Q(1)
2882        INTEGER     IQ(1)
2883        CHARACTER*1 CQ(1)
2884        LOGICAL LQ(1)
2885        COMMON /ARRAY/  Q, CQ
2886        COMMON /MEMORY/ MAXC, MAXQ
2887        EQUIVALENCE(IQ,Q)
2888        COMMON /QACOMMON/ PROGRM, PVERSN, PDATE,
2889       &   RUNDAT, RUNTIM, AUTHOR, SPONSR
2890        CHARACTER*12 PROGRM
2891        CHARACTER*8 PVERSN, PDATE
2892        CHARACTER*8 RUNDAT, RUNTIM
2893        CHARACTER*80 AUTHOR, SPONSR
2894
2895        DATA PROGRM / '            ' /
2896        DATA PVERSN / '        ' /
2897        DATA PDATE  / '        ' /
2898        DATA RUNDAT / '        ' /
2899        DATA RUNTIM / '        ' /
2900        DATA AUTHOR / ' ' /
2901        DATA SPONSR / ' ' /
2902
2903  C   --   PROGRM - the program name
2904  C   --   PVERSN - the program version number
2905  C   --   PDATE - the program revision date
2906  C   --   RUNDAT, RUNTIM - the run date and time
2907  C   --   AUTHOR - the program author
2908  C   --   SPONSR - the program sponsor
2909  *COMDECK DYNAME
2910  C*********************************************************************
2911  C       /DYNAME/ ($include 'DYNAME.INC')
2912  C        CNAMES = Names of CHARACTER dynamic arrays (CHARACTER*8)
2913  C        DTYPES = Types of REAL dynamic arrays (CHARACTER*8)
2914  C        RNAMES = Names of REAL data dynamic arrays (CHARACTER*8)
2915  C*********************************************************************
2916  C234567
2917        CHARACTER*(8) CNAMES(3), DTYPES(4), RNAMES(4)
2918        COMMON /DYNAME/ CNAMES, DTYPES, RNAMES
2919  C*********************************************************************
2920
```

```
2921
2922  *COMDECK FFRDAT
2923  C*********************************************************************
2924  C     NFORM  = "            digits or characters contained in a word
2925  C     NSIZE  = "            words per record
2926  C     NCOL   = Max. no. of Characters per Record (record length)
2927  C*********************************************************************
2928        INTEGER MFIELD, NFORM, NCOL
2929        PARAMETER( NFORM = 80, MFIELD= 100, NCOL = 80 )
2930  *COMDECK GEOPAR
2931  C**********************************************************************
2932  C    /GEOPAR/ ($include 'GEOPAR.INC')
2933  C     HEADER = Character description of problem, etc...
2934  C     IMAX   = No. of I-lines in grid
2935  C     JMAX   = No. of J-lines in grid
2936  C     KMAX   = No. of K-lines in grid
2937  C     LESSEL = List of Element Side sets concatenated Element List
2938  C     LESSNL = List of Element Side sets concatenated Nodes associated
2939  C              with each element side set Node List
2940  C     LNPSNL = List of Nodal Point set concatenated Nodes List
2941  C     MAXNENO= Max. no. of nodes of any element
2942  C     NDIM   = Number of physical DIMensions
2943  C     NDBCAM = CAMDAT versoin number
2944  C     NELBLK = Number of ELement BLocKs in mesh
2945  C     NELX   = No. of elements along X-axis of mesh
2946  C     NELY   = No. of elements along Y-axis of mesh
2947  C     NELZ   = No. of elements along Z-axis of mesh
2948  C     NQAREC = Number of QA records
2949  C     NSTEP  = No. of time steps in input CAMDAT (INTEGER)
2950  C     NUMEL  = NUmber of ELeMents in mesh
2951  C     NUMNOD = NUMber of NODes in mesh
2952  C     NUMNPS = No. of Nodal Point Sets
2953  C     NUMESS = No. of Element Side Sets
2954  C     NUQATR = No. of UniQue ATtRibute names
2955  C     NUQPR0 = No. of UniQUe PRoperty names (old)
2956  C     NUQPRP = No. of UniQUe PRoPerty names (new)
2957  C     NVAREL = No. of ELEMENT type ANALYSIS variables
2958  C     NXINFO = No. of information_records
2959  C**********************************************************************
2960  C234567
2961        CHARACTER*(80) HEADER
2962        INTEGER
2963       1IMAX,    JMAX,    KMAX,    MAXNENO, LESSEL,  LESSNL,  LNPSNL,
2964       2NDBCAM,  NDIM,    NELBLK,  NELX,    NELY,    NELZ,    NQAREC,
2965       3NSTEP,   NUMEL,   NUMESS,  NUMNOD,  NUMNPS,  NUQATR,  NUQPR0,
2966       4NUQPRP,  NVARHI,  NVARGL,  NVARND,  NVAREL,  NXINFO
2967        COMMON /GEOPAR/
2968       1IMAX,    JMAX,    KMAX,    MAXNENO, LESSEL,  LESSNL,  LNPSNL,
2969       2NDBCAM,  NDIM,    NELBLK,  NELX,    NELY,    NELZ,    NQAREC,
2970       3NSTEP,   NUMEL,   NUMESS,  NUMNOD,  NUMNPS,  NUQATR,  NUQPR0,
2971       4NUQPRP,  NVARHI,  NVARGL,  NVARND,  NVAREL,  NXINFO,  HEADER
2972  C**********************************************************************
2973
```

```
2974
2975  *COMDECK INDEX
2976  C*********************************************************************
2977  C     /INDEX/ ($include 'INDEX.INC')
2978  C      LASCIDX= Last index of CHARACTER*1 dynamic arrays (INTEGER)
2979  C      BCDAIDX= First index of current CHAR*1 dynamic array (INTEGER)
2980  C      ECDAIDX= Last  index of current CHAR*1 dynamic array (INTEGER)
2981  C      LASRIDX= Last index of REAL dynamic arrays (INTEGER)
2982  C      BRDAIDX= First index of current REAL dynamic array (INTEGER)
2983  C      ERDAIDX= Last  index of current REAL dynamic array (INTEGER)
2984  C*********************************************************************
2985  C234567
2986        INTEGER BCDAIDX,BRDAIDX,ECDAIDX,ERDAIDX,LASCIDX,LASRIDX
2987        COMMON /INDEX/ BCDAIDX,BRDAIDX,ECDAIDX,ERDAIDX,LASCIDX,LASRIDX
2988  C*********************************************************************
```

```
2989
2990  *COMDECK INFIL
2991  C  INFIL.INC
2992  C*****************************************************************
2993  C
2994  C234567
2995        REAL BARSOIL1, BARSOIL2, ETFACT, IMBFACT, PPTFACT, RKPOR,
2996       +     RKMMFACT, SKSFACT,  SDFACT, SNOPAR1, SUBPAR1
2997        CHARACTER*8 REGION, CDB_NAME(MAXVAR), INFIL_NAME(MAXVAR)
2998        CHARACTER TITLE1*80, TITLE2*40, RUNDATE*8, RUNTIME*8
2999  C*      LOGICAL
3000  C
3001        COMMON /INFILNAM/ CDB_NAME, INFIL_NAME, REGION
3002        COMMON /INFILDAT/ BARSOIL1, BARSOIL2, ETFACT, IMBFACT, PPTFACT,
3003       +                  RKPOR, RKMMFACT, SKSFACT, SDFACT, SNOPAR1,
3004       +                  SUBPAR1
3005        DATA INFIL_NAME/'BARSOIL1','BARSOIL2','ETFACT  ','IMBFACT ',
3006       +              'PPTFACT ','RKPOR   ','RKMMFACT','SKSFACT ',
3007       +              'SDFACT  ','SNOPAR1 ','SUBPAR1 ',89*'        '/
3008  C*****************************************************************
3009
```

```
*COMDECK IOCOM
      INTEGER
     +ITEMP1,ICDB,ISCREN,NOUTFL,INASCI,INPTMP,INPINF
      COMMON /IOCOM/
     +ITEMP1,ICDB,ISCREN,NOUTFL,INASCI,INPTMP,INPINF
```

```
*COMDECK LENGTH
C*********************************************************************
C     /LENGTH/ ($include 'LENGTH.INC')
C      CLENGT = Length of CHARACTER*1 base type dynamic arrays
C      DLENGT = Length of Data types of dynamic arrays
C      RLENGT = Length of REAL base type dynamic arrays
C*********************************************************************
C234567
      INTEGER CLENGT(3), DLENGT(4), RLENGT(4)
      COMMON /LENGTH/ CLENGT, DLENGT, RLENGT
C*********************************************************************
```

```
3027  *COMDECK LOCAT
3028  C*********************************************************************
3029  C     /LOCATC/ ($include 'LOCAT.INC')
3030  C      LCPTR  = Array used for pointing to each dynamic array start
3031  C               location in the base CHARACTER*1 array, C.
3032  C      L[name]= Pointer to CHAR dynamic array ( L[dynamic-array-name] )
3033  C*********************************************************************
3034  C     /LOCATR/ ($include 'LOCAT.INC')
3035  C      LRPTR  = Array used for pointing to each dynamic array start
3036  C               location in the base REAL array, Q.
3037  C      L[name]= Pointer to REAL dynamic array ( L[dynamic-array-name] )
3038  C*********************************************************************
3039  C234567
3040        INTEGER
3041       $LIDEBLK, LNUMPRP,  LIASPRP,  LXMATPR,
3042       $LRPTR(4)
3043        EQUIVALENCE (LRPTR,LIDEBLK)
3044        COMMON /LOCATR/
3045       $LIDEBLK, LNUMPRP,  LIASPRP,  LXMATPR
3046        INTEGER
3047       $LQAINFO,  LNAMELB, LNMATPR, LCPTR(3)
3048        EQUIVALENCE (LCPTR,LQAINFO)
3049        COMMON /LOCATC/
3050       $LQAINFO,  LNAMELB, LNMATPR
3051  C*********************************************************************
```

```
*COMDECK MISCDAT
C*********************************************************************
C     /MISCDAT/ (from FORTRAN statement: INCLUDE 'MISCDAT.INC')
C      LTYEARS = Logical variable,
C              =.TRUE.; Time variables used are to be in YEARS
C               (Convert intrusion times specified in SECONDS
C               to YEARS)
C      LTSECON = Logical variable,
C              = .TRUE.; Time variables used are to be in SECONDS
C                (Convert intrusion times specified in YEARS to
C                 SECONDS)
C      GLOBAL
C      DEBUG   = Logical variable,
C              = .TRUE.; Write debug statements to debug file (NOUTFL)
C*********************************************************************
C234567
      INTEGER HIFLAG
      LOGICAL DEBUG, GLOBAL, LTSECON, LTYEARS
      COMMON /MISCDAT/ HIFLAG, DEBUG, GLOBAL, LTSECON, LTYEARS
C*********************************************************************
```

```
*COMDECK NEWPAR
      INTEGER NUMVAR
      CHARACTER*(8) NAMVAR(MAXVAR)
      CHARACTER*(20) FILEMAIN, FILEECHO, FILESUMM
      COMMON /NEWPAR/ NUMVAR, NAMVAR, FILEMAIN, FILEECHO, FILESUMM
```

```
*COMDECK PARACM
C*********************************************************************
C     MAXVAR = Max. no. of NEW PREINFIL sampled or modified fixed variables
C     MAXREC = Max. no. of records in template control file
C     MAXDAT = Max. no. of (unmodified) data items in INFIL UTF template
C*********************************************************************
      INTEGER MAXDAT, MAXREC, MAXVAR
      PARAMETER( MAXDAT = 1000, MAXREC=1000, MAXVAR = 100 )
      EQUIVALENCE(LQ,Q)
```

ATTACHMENT V

Documentation of POSTINFIL (to satisfy AP-SI.1Q)

This attachment and attachment IV contains documentation necessary to satisfy procedure AP-SI.1Q, Section 5.1.1.2 verification of routines used. POSTINFIL can be considered to be a routine since it can be verified by visual inspection and hand calculations. Its purpose is to create automatically and readily input user control files for the computer model INFIL as used in these analyses. Attachment VI contains a listing of the source code for POSTINFIL.

POSTINFIL Version A2.50 Software Routine Verification

TRANSFERS INFILTRATION RATE MAP AND NUMBER OF ACTIVE CELLS IN MAP OR SUMMARY INFILTRATION RATE OF APPROPRIATE POSTINFIL OUTPUT FILES TO COMPUTATIONAL DATABASE (CDB) FILES ACCORDING TO DIRECTIVES SPECIFIED IN A USER INPUT CONTROL FILE. (AUTHOR: RON MCCURLEY, SNL-PAO)

Documentation to satisfy first and second bullet of Section 5.1.1.2:

A software routine (consisting of several modules) was developed in accordance with AP-SI.1Q for the purpose of automatically and readily transferring infiltration rate map data and the number of active cells in that map, or summary infiltration rate from appropriate output files of INFIL V A_2.a1 to which also incorporate 1) selected parameter sampled values (using Latin Hypercube Sampling) and 2) designated filenames which are read from this file and output by POSTINFIL. The software developed, POSTINFIL, version A2.50, is in FORTRAN 77. The source code and executable reside in the following directory location on a DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories:

I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.SOURCE_CODE].

A listing of the source code is included in Attachment VI.

This satisfies the requirements of Section 5.1.1.2, first and second bullets of AP-SI.1Q for routines.

Documentation to satisfy third bullet of Section 5.1.1.2:

The correct and accurate transfer of spatially averaged infiltration rates and corresponding values of active grid cells into each output CDB file produced by POSTINFIL is verified by executing POSTINFIL using the input user control file and input CDB file (from postlhs) and observing that 1) the POSTINFIL code terminates normally, 2) the correct values of spatially averaged infiltration rates and corresponding values of active grid cells are placed into the output CDB files. POSTINFIL requires the following input; 1) a (binary) 'computational' database file (CDB), 2) a user control file which contains instructions for POSTINFIL. The output from POSTINFIL consists of new CDB files, satisfying requirement 2) above, and an optional debug file. A sample of the input user control file is included Table VII-10. To examine the output CDB files, there is a software tool called GROPECDB. GROPECDB is used only for display of data contained in a binary file (the CDB file) and is exempt from AP-SI.1Q. An ascii output printout of the CDB file (corresponding to realization 1) using GROPECDB is shown in Table V-1. This file shows the spatially averaged infiltration rate (6.48678E+01) for the active grid cells. To verify that the calculated spatial average as calculated by POSTINFIL is correct, a comparison is made between this value as found in the CDB and as calculated by a hand

calculator using the grid cells within the specific rectangular region as modeled in this AMR for realization '001' of the 'se10' watershed for the glacial transition climate. The output file from INFIL displaying infiltration rates per cell is shown in Table V-2. The value in the CDB file as seen in the GROPECDB listing is 6.48678E+01. The value calculated with a hand calculator for the 8 active cells (in bold in Table V-2) with the rectangular region is 6.48678E+01. Since this is an arbitrary selection, it is a reasonable verification of this capability of the code POSTINFIL.

Table V-1 GROPECDB Display of Output Infiltration Rates from INFIL

```
GROPECDB_PA96

 GGGGG  RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB         PPPPPP
GG  GGG RRRRR   OO   OO PP      EE      CC      DD   DD BB   BB         PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB         PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB  _______ PP

                      GROPECDB_PA96 Version 2.12
                       PROD PA96 Built 06/27/96
                       Sponsored by Amy Gilkey

                     Run on 01/25/00 at 15:18:45
                  Run on ALPHA AXP BEATLE OpenVMS V7.1

Database:  I1:[000000.RDMCCUR.YMP99.INFILTRATION.POSTINFIL.GLACIAL]POSTINF_WW1_R001.CDB;3
Written on:  11/29/99 12:45:44

CAMDAT Version:   1        (EXODUS Version:   1)

GLOBAL TIME STEP VARIABLES

   Step        Time        NCELLSE1       INF_SE1        RUN_SE1        NCELLSE2
                           INF_SE2        RUN_SE2        NCELLSE3       INF_SE3
                           RUN_SE3        NCELLSE4       INF_SE4        RUN_SE4
                           NCELLSE5       INF_SE5        RUN_SE5        NCELLSE6
                           INF_SE6        RUN_SE6        NCELLSE7       INF_SE7
                           RUN_SE7        NCELLSE8       INF_SE8        RUN_SE8
                           NCELLSE9       INF_SE9        RUN_SE9        NCELSE10
                           INF_SE10       RUN_SE10       NCELLWT2       INF_WT2
                           RUN_WT2        NCELLDH4       INF_DH4        RUN_DH4
                           NCELLDH3       INF_DH3        RUN_DH3        NCELLCW1
                           INF_CW1        RUN_CW1        NCELLSW1       INF_SW1
                           RUN_SW1        NCELLWW1       INF_WW1        RUN_WW1
      1   0.00000E+00    6.10000E+01    1.00936E+02    0.00000E+00    5.00000E+00
                         8.30917E+01    0.00000E+00    4.70000E+01    1.23239E+02
                         0.00000E+00    8.20000E+01    1.18034E+02    0.00000E+00
                         4.10000E+01    1.00655E+02    0.00000E+00    1.14000E+02
                         1.08497E+02    0.00000E+00    6.70000E+01    9.07320E+01
                         0.00000E+00    7.00000E+01    9.58903E+01    0.00000E+00
                         8.50000E+01    1.01568E+02    0.00000E+00    8.00000E+00
                         6.48678E+01    0.00000E+00    1.58300E+03    5.93866E+01
                         0.00000E+00    6.10000E+02    6.01417E+01    0.00000E+00
                         1.18800E+03    6.04595E+01    0.00000E+00    4.17000E+02
                         6.94956E+01    0.00000E+00    7.18000E+02    6.40361E+01
                         0.00000E+00    8.75000E+02    6.31565E+01    0.00000E+00
```

Table V-2. Output From INFIL Showing Selected Variable Values Including Net Annual Infiltration Rates By Grid Cell

easting	northing	POSTclp	rain	snow-fall	snow-cover	snow-melt	sublimation	evapotrans	run-infil	del-soil	net-infil	runoff	run-on	mass-balance	max-balance	mass-balance #2
547451.0	4078833.0	335.23593	262.58986	72.64607	876.91791	66.00954	5.77788	234.51203	0.00000	4.55407	89.53330	0.00000	0.000000	-0.1548983E-11	-0.4940492E-13	0.4405365E-12
547451.0	4078803.0	335.12734	262.50480	72.62254	871.25021	65.79229	5.97386	230.29920	0.00000	4.54790	93.44999	0.00000	0.000000	-0.1804779E-11	-0.5329071E-13	0.5258016E-12
547421.0	4078803.0	333.82920	266.29874	67.53045	775.77963	61.07342	5.61248	251.36128	0.00000	4.58303	71.42785	0.00000	0.000000	0.7105427E-13	-0.3794187E-13	-0.1278977E-12
547421.0	4078773.0	333.82920	266.29874	67.53045	775.77983	61.07352	5.61239	251.36039	0.00000	4.58304	71.42883	0.00000	0.000000	-0.1065814E-12	-0.3802514E-13	-0.1278977E-12
547391.0	4078773.0	332.96932	265.75405	67.21528	755.41416	60.79408	5.58459	263.85231	0.00000	4.48366	58.21216	0.00000	0.000000	-0.4760636E-12	-0.3907985E-13	-0.1705303E-12
547391.0	4078743.0	332.86218	265.66853	67.19365	751.28995	60.64830	5.71066	264.88062	0.00000	4.46021	56.97600	0.00000	0.000000	-0.3481659E-12	0.4429790E-13	0.1421085E-13
547361.0	4078743.0	332.22062	265.15648	67.06414	736.45340	60.54575	5.68987	280.11907	0.00000	4.40810	39.01341	2.16165	0.000000	-0.8242296E-12	-0.8215650E-13	-0.1989520E-12
547361.0	4078713.0	331.90076	264.90119	66.99957	729.26860	60.50165	5.67245	279.96899	0.00000	4.40464	38.90102	2.10819	0.000000	-0.3907985E-12	-0.8959500E-13	-0.2984279E-12
547331.0	4078713.0	331.15679	264.30740	66.84939	713.88551	60.42643	5.60450	275.89926	2.16165	4.09794	46.89828	0.00000	0.000000	-0.1527667E-11	-0.4268808E-13	-0.2415845E-12
547331.0	4078683.0	331.15679	264.30740	66.84939	715.24030	60.55110	5.47904	274.44367	2.10819	4.12288	48.40013	0.00000	0.000000	-0.1080025E-11	-0.4629630E-13	-0.2415845E-12
547301.0	4078683.0	330.31077	268.65535	61.65541	638.35519	55.66799	5.17615	322.48432	0.00000	1.83902	0.00000	0.00000	0.000000	-0.1421085E-12	0.1003642E-12	0.1563194E-12
547271.0	4078653.0	329.67907	268.14157	61.53750	626.62878	55.69172	5.03982	322.18628	0.00000	1.64701	0.00000	0.00000	0.000000	0.2202682E-12	-0.1136868E-12	-0.1918465E-12
547241.0	4078623.0	329.04983	267.62978	61.42005	613.88945	55.55046	5.06984	324.01997	0.00000	-0.83973	0.00000	0.00000	0.000000	0.2273737E-12	-0.9392487E-13	0.3055334E-12
547211.0	4078593.0	328.52727	270.80057	57.72670	590.61387	51.92048	5.01148	328.00809	0.00000	-5.28704	0.00000	0.00000	0.000000	-0.1563194E-12	-0.1509903E-12	-0.6394885E-13
547181.0	4078563.0	327.90248	270.28557	57.61691	577.94201	51.88177	4.94619	327.48242	0.00000	-5.31508	0.00000	0.00000	0.000000	-0.6039613E-12	0.1483258E-12	-0.2344791E-12
547151.0	4078533.0	327.59099	270.02881	57.56218	571.60049	51.86920	4.90702	320.95966	0.00000	0.93835	0.00000	0.00000	0.000000	-0.5968559E-12	-0.1139089E-12	-0.1776357E-12
547121.0	4078503.0	327.17655	269.68719	57.48936	562.83232	51.78948	4.91821	320.59254	0.00000	0.88413	0.00000	0.00000	0.000000	-0.2771117E-12	-0.9503509E-13	-0.3694822E-12

ATTACHMENT VI

POSTINFIL SOURCE CODE

This attachment contains a listing of the software routine POSTINFIL developed at Sandia National Laboratories which serves as a postprocessor to transfer output from INFIL V. A_2.a1 to a CDB file. Verification of the use of this routine is found in Attachment V.

Post-processor POSTINFIL V. 2.50 for the INFIL V. A_2.a1 code. See flowchart entitled FLWCHR.INC located near the end of this file.

```
*DECK A1MAIN
      PROGRAM POSTINFIL
C***********************************************************************
C
C                                POSTINFIL
C                                ----
C
C                      A  post-processor for INFIL
C
C                             August 1999
C
C                               Author
C                               ------
C
C                        Ron D McCurley
C                      Information Systems Division
C                NEW MEXICO ENGINEERING RESEARCH INSTITUTE
C                       University of New Mexico
C                        851 University Blvd SE
C                               Suite 101
C                      Albuquerque, NM  87106-4342
C                            (505) 272-7227
C
C                               Abstract
C                               --------
C
C  POSTINFIL reads the output from INFIL
C  It also reads the input CDB file
C  (CAMDAT Data Base),
C  assembles the infitrations according to spatial discertization and
C  select times. It finally writes time dependent infiltration maps
C  to an output CDB file
C
C                              Limitations
C                              -----------
C
C
C
C                         Primary Reference
C                         -----------------
C
C                        Secondary References
C                        --------------------
C
C                             Update History
C                             --------------
C
C  VERSION     DATE          MODIFIER        CHANGES
C  A1.00       22-JAN-1999   R.D.McCurley    -Created for 99 TSPA
```

```
C  A1.10      12-AUG-1999   R.D.McCurley    -Modified for new version of
C                                             INFIL(2A)
C  A2.00      19-AUG-1999   R.D.McCurley    -Modified for adding processing
C                                             of INFIL summary output
C  A2.10      01-SEPT-1999  R.D.McCurley    -Modified for adding processing
C                                             of INFIL summary output
C                                             variable "RUNOFF"
C  A2.20      22-SEPT-1999  R.D.McCurley    -Modified for adding processing
C                                             of INFIL summary output
C                                             variable "NUMCELL"
C  A2.50      05-OCT-1999   R.D.McCurley    -Modified for adding processing
C                                             of POSTINFIL input file
C
C
C                              Disclaimer
C                              ----------
C
C     This computer program was prepared as an account of work
C  sponsored by an agency of the United States Government. Neither
C  the United States Government nor any agency thereof, nor any of
C  their employees, nor any of their contractors, subcontractors,
C  or their employees, makes any warranty, express or implied, or
C  assumes any legal liability or responsibility for the accuracy,
C  completeness, or usefulness of any information, apparatus,
C  product, or process disclosed, or represents that its use would
C  not infringe privately owned rights. Reference herein to any
C  specific commercial product, process, or service by trade name,
C  trademark, manufacturer, or otherwise, does not necessarily
C  constitute or imply its endorsement, recommendation, or
C  favoring by the United States Government, any agency thereof
C  or any of their contractors or subcontractors. The views and
C  opinions expressed herein do not necessarily state or reflect
C  those of the United States Government, any agency thereof or
C  any of their contractors or subcontractors.
C
C
C                             Program flowchart
C                             -----------------
C
      INCLUDE 'FLWCHR.INC'
C
C
C                        Assumptions and Limitations
C                        ---------------------------
C
C    Language used is ANSI X3.9-1978 FORTRAN 77 except that
C    comments and Hollerith strings use lowercase characters.
C    INTEGER and REAL variable names are explicitly typed.
C    Machine dependent coding exists in subroutines: QAPAGE
C
C                              Statistics
C                              ----------
C
C     Version:      A1.20                language:        FORTRAN 77
C     Computer:     ALPHA AXP            Total Lines:
C     Date:         08/12/99             Code Lines:
C                                        Percent Comment:
```

```
107   C
108   C
109   C                          Types of data sets
110   C                          ------------------
111   C
112   C  INPUT/
113   C    BINARY: in units 'IDB'
114   C    ASCII:  in units 'INPUTF','ISCRAT'
115   C
116   C  OUTPUT/
117   C    BINARY: in units 'ODB'
118   C    ASCII:  in units 'NOUTFL'
119   C
120   C                              Files used
121   C                              ----------
122   C
123   C  filename        unit  description
124   C  <PREPOSTINFIL.CDB>  30    Input Computational Data Base (REQUIRED)
125   C  <POSTINFIL.INP>      1    Input text file (REQUIRED)
126   C  <INFILPT.OUT>        9    Input INFIL pointwise infiltrations (OPTIONAL)
127   C  <INFILSUM.OUT>      10    Input INFIL summary infiltrations (REQUIRED)
128   C  <POSTINFIL.CDB>     31    Output Computational Data Base (REQUIRED)
129   C  <POSTINFIL.DBG>      6    Output diagnostics/debug file (OPTIONAL)
130   C
131   C*********************************************************************
132   C234567
133         IMPLICIT NONE
134         INCLUDE 'LOCAT.INC'
135         INCLUDE 'GEOPAR.INC'
136         INCLUDE 'PARAMS.INC'
137         INCLUDE 'MISCDAT.INC'
138         INCLUDE 'FILEUN.INC'
139         INCLUDE 'QA.INC'
140         INCLUDE 'TOOLS.INC'
141         INCLUDE 'POSTINFIL.INC'
142
143         INTEGER  IERR, MAXC, MAXQ, MORMEMC, MORMEMR
144         INTEGER IQ(1)
145         REAL CPUSEC
146         REAL Q(1)
147         REAL INFIL_AVE, RUNOFF_AVE
148         REAL PRECIP(MAXSOLN),ET(MAXSOLN),RUNONINF(MAXSOLN),
149        $     NET_INF(MAXSOLN),RUNOFF(MAXSOLN),RUNON(MAXSOLN),
150        $     MEANH2OERR(MAXSOLN),MAXH2OERR(MAXSOLN),INFPT(MAXSOLN)
151         REAL TIME( MAXTIMES )
152         CHARACTER*(1) C(1)
153         CHARACTER*(8) BLKNAM, NAMGLV2(3)
154         CHARACTER*(8) TUNITS
155         CHARACTER*(132) FIDB, FODB, FTRN, FTRN2, FTXT
156
157         LOGICAL LQ(1)
158         EQUIVALENCE( IQ, Q )
159         EQUIVALENCE( LQ, Q )
160         EXTERNAL BLOCK
161   C
162   C<><><><><><><><><><><><>
163   C...BEGIN PROCEDURES...
```

```
164   C<><><><><><><><><><><><>
165   C
166   C     ...Set-up constants
167         MAXEAST = 548690.0
168         MAXNORTH = 4081200.0
169         MINEAST = 547350.0
170         MINNORTH = 4076700.0
171         CALL SUCONS
172
173   C     ...Prompt user for PROGRAM control options
174         CALL PREPRO( FIDB, FTXT, FTRN, FTRN2, FODB )
175
176   C     ...Initialize DYNAMIC base arrays
177         CALL MDINIT( Q )
178         CALL MCINIT( C )
179
180   C     ...Set-up the DYNAMIC base arrays
181         IERR = 0
182         CALL DBSETUP( Q, C, IERR )
183         IF(IERR.NE.0)CALL QAABORT('(A1MAIN) - Setting up DYNAMIC base'//
184        $                          ' arrays %%%')
185
186   C     ...Set debugging/diagnostics file unit
187         CALL DBERRUNI( NOUTFL )
188
189   C     ...Process input CAMDAT
190         CALL DBIOPEN( IDB, FIDB, Q, C, IERR )
191         IF(IERR.NE.0)CALL QAABORT('(A1MAIN) - Opening input CAMDAT %%%')
192         CALL PROCDB( MAXC, MAXQ, IQ, LQ, Q, C )
193   C
194   C     ...Process POSTINFIL input user control file
195         CALL FILOPEN( ICNTRLFIL, 'INP', 'FORM', FTXT, IERR )
196         IF(IERR.NE.0) CALL QAABORT('(A1MAIN) - opening POSTINFIL'//
197        $                          ' input text file %%%')
198         CALL PROCIN(NAMGLV2)
199
200         IF (FTRN(1:6) .NE. 'CANCEL' .AND. FTRN(1:1) .NE. ' ') THEN
201            SUBSHED = .TRUE.
202            CALL FILOPEN( INFLOUT1, 'INP', 'FORM', FTRN, IERR )
203            IF(IERR.NE.0)CALL QAABORT('(A1MAIN) - opening INFIL'//
204        $                            ' pointwise output text file %%%')
205   C  UTMEAST      = utm easting [m]
206   C  UTMNORTH     = utm northing [m]
207   C  PRECIP       = average annual precip [mm/yr]
208   C  ET           = average annual evapotranspiration [mm/yr]
209   C  RUNONINF     = average annual infiltration from runon [mm/yr]
210   C  ROOTZONE_H2O = average annual change in root-zone water content [mm/yr]
211   C  NET_INF      = average annual net infiltration rate [mm/yr]
212   C  RUNOFF       = average annual runoff generation [mm/yr]
213   C  RUNON        = average annual surface water runon [mm/yr]
214   C  MEANH2OERR   = average annual water balance calculation error [mm/yr]
215   C  MAXH2OERR    = maximum daily water balance calculation error  [mm/day]
216            CALL PRCOUT( PRECIP,ET,RUNONINF,NET_INF,RUNOFF,RUNON,MEANH2OERR,
217        $                MAXH2OERR )
218
219            CLOSE(INFLOUT1)
220         ENDIF
```

```
221   C
222         IF (FTRN2(1:6) .NE. 'CANCEL' .AND. FTRN2(1:1) .NE. ' ') THEN
223           CALL FILOPEN( INFLOUT2, 'INP', 'FORM', FTRN2, IERR )
224           IF(IERR.NE.0)CALL QAABORT('(A1MAIN) - opening INFIL'//
225        $                            ' summary output text file %%%')
226
227            CALL PRCOUT2(INFLOUT2,NUMCELL,INFIL_AVE,RUNOFF_AVE)
228            CLOSE(INFLOUT2)
229         ENDIF
230   C     ...Open scratch file and output CAMDAT
231         CALL DBOOPEN( ODB, IDB, IDBSCR, FODB, IERR )
232         IF(IERR.NE.0)CALL QAABORT('(A1MAIN) - Opening output'//
233        $                          ' CAMDAT and SCRATCH files %%%')
234
235   C     ...Assemble QA record for program POSTINFIL
236         CALL QAMAKREC( NQAREC, C(LQAINFO) )
237   C
238         CALL ADDMEM1(  MORMEMR, MORMEMC )
239         CALL DYNMEM(  MORMEMR, MAXQ, Q )
240         CALL ZERODA( IQ, LQ )
241         CALL DYNMEMC( MORMEMC, MAXC, C )
242
243   C     ...Write POSTINFIL output CAMDAT file header section
244         CALL WRCHEA( Q, C, C(LQAINFO), C(LNAMELB), C(LSAVNAM) )
245   C
246         TIME(1) = 0.0
247   C     ...Write output infiltration maps to output CDB
248         CALL WRTANA (Q,       C,       TIME,    C(LNAMELB),  NAMGLV2,
249        $             INFIL_AVE, INFPT, NET_INF, RUNOFF_AVE )
250   C
251   C**** Perform ending procedures ****
252   C
253         WRITE(NOUTFL,900)
254         CALL DBPTITLE(ODB,NOUTFL,HEADER)
255         CALL DBPVERS(ODB,NOUTFL)
256         CALL DBPQAREC(ODB,NOUTFL,NQAREC,C(LQAINFO))
257         CALL DBPSIZES(ODB,NOUTFL)
258         CALL DBICLOSE(IDB,IERR)
259         IF(IERR.NE.0)CALL QAABORT('(A1MAIN) - Closing input CAMDAT %%%')
260         CALL DBOCLOSE(ODB,IERR)
261         IF(IERR.NE.0)CALL QAABORT('(A1MAIN) - Trouble closing output'//
262        $' cdb-scratch & CAMDAT files, etc... %%%')
263         CALL QACPUS(NOUTFL,CPUSEC)
264         CALL QAPAGE(NOUTFL,'END')
265         CLOSE(ISCRAT)
266         CLOSE(NOUTFL)
267
268         STOP 'POSTINFIL Normal Completion'
269   C
270   C--------------- F O R M A T   S T A T E M E N T S -----------------
271   C
272    900  FORMAT(/,1X,72('*'),/,1X,'(A1MAIN)',//)
273   C
274   C-----------------------------------------------------------------------
275   C**** END OF PROGRAM POSTINFIL (A1MAIN) ****
276   C-----------------------------------------------------------------------
277   C
```

```
      END
*DECK ADDMEM
      SUBROUTINE ADDMEM( MORMEMR, MORMEMC )
C*********************************************************************
C***                                                               ****
C***        computes  A_D_D_itional  M_E_M_ory  needed  module     ****
C***                  - - -          - - -                         ****
C*********************************************************************
C
C  PURPOSE:       Sets dynamic array lengths, and computes the
C                 amount of memory needed for the REAL base
C                 Q array and CHARACTER*1 base C array.
C
C  AUTHOR:        Jonathan Scott Rath, Esq. [NMERI]
C
C  UPDATED:       17-AUG-1995  --JSR: Deleted TDEC array
C                 26-AUG-1994  --JSR: Adapted for new GENERIC.RDB
C                 14-JUN-1994  --JSR: First Ed. for POSTINFIL.
C
C  CALLED BY:     PROCDB
C                 PROGEN
C                 POSTINFIL {main program}
C
C  ENTRY POINTS: ADDMEM1
C                ADDMEM2
C
C  ARGUMENTS:
C  ENTRY/
C    --common blocks
C     /FILEUN/ ($include 'FILEUN.INC')
C      NOUTFL = File unit of diagnostics/debug output file
C
C     /DYNAME/ ($include 'DYNAME.INC')
C      CNAMES = Names of the CHARACTER*1 type dynamic arrays
C      DTYPES = Data types of the CNAMES arrays
C      RNAMES = Names of the REAL type dynamic arrays
C
C     /GEOPAR/ ($include 'GEOPAR.INC')
C      NELBLK = Number of ELement BLocKs in mesh (materials)
C      NQAREC = Number of QA records
C      NUQPRP = No. of UniQUe PRoperty names
C
C     /MISCDAT/ (from FORTRAN statement: INCLUDE 'MISCDAT.INC')
C      NSOLNS  = No. of times [INTEGER]
C      NUMCELL    = No. of cells used by POSTINFIL [INTEGER]
C      NORIAC  = No. Of Radioisotopes In All Chains [INTEGER]
C
C     /TOOLS/ ($include 'TOOLS.INC')
C      DEBUG  = Logical for writing debugging information to .dbg file
C      WRTHIS = Logical for writing Radioisotope Inventory
C               History table file.
C
C  LOCAL/
C      IERR   = INTEGER error flag
C
C  EXIT/
C    --common block
```

```
335   C      /LENGTH/ ($include 'LENGTH.INC')
336   C       CLENGT = CHARACTER*1 type dynamic array lengths array
337   C       RLENGT = REAL type dynamic array LENGTHS array
338   C
339   C      /INDEX/ ($include 'INDEX.INC')
340   C       LASCIDX = Last CHARACTER dynamic array index
341   C       BCDAIDX = Beginning index of CHARACTER dyanamic arrays
342   C       ECDAIDX = Ending "                                "
343   C       LASRIDX = Last REAL dynamic array index
344   C       BRDAIDX = Beginning index of REAL dyanamic arrays
345   C       ERDAIDX = Ending "                                "
346   C
347   C     --through subroutine call
348   C       MORMEMR= No. of REAL words of memory needed
349   C       MORMEMC= No. of CHARACTER*1 words of memory needed
350   C
351   C*********************************************************************
352   C234567
353         IMPLICIT NONE
354         INCLUDE 'FILEUN.INC'
355         INCLUDE 'DYNAME.INC'
356         INCLUDE 'GEOPAR.INC'
357         INCLUDE 'LENGTH.INC'
358         INCLUDE 'INDEX.INC'
359         INCLUDE 'MISCDAT.INC'
360   C*      INCLUDE 'POSTINFIL.INC'
361         INCLUDE 'TOOLS.INC'
362         INTEGER I, IERR, MORMEMC, MORMEMR
363   C
364   C<><><><><><><><><><><><>
365   C...Begin Procedures...
366   C<><><><><><><><><><><><>
367   C
368   C*********************************************************************
369   C
370   C     PSEUDO ENTRY ADDMEM( MORMEMR, MORMEMC )
371   C
372   C*********************************************************************
373   C
374   C     THIS SECTION COMPUTES MEMORY REQUIRED FOR THE ARRAYS ASSOCIATED
375   C     WITH THE INPUT CAMDAT.
376   C
377   C     (CALLED BY PROCDB)
378   C
379         IERR       = 0
380         MORMEMC    = 0
381         MORMEMR    = 0
382   C     ...Set beginning and ending dynamic array indices
383         BRDAIDX    = 1
384         ERDAIDX    = 4
385         BCDAIDX    = 1
386         ECDAIDX    = 3
387   C     ...Set dynamic REAL array lengths
388   C     ...(1:4) IDEBLK, NUMPRP, IASPRP, XMATPR
389         RLENGT(1)  = NELBLK
390         RLENGT(2)  = RLENGT(1)
391         RLENGT(3)  = NELBLK*NUQPRP
```

```
      RLENGT(4)  = RLENGT(3)
C     ...Set dynamic CHARACTER*1 array lengths
C     ...(1:3) QAINFO, NAMELB, NMATPR
      CLENGT(1) = (NQAREC+2)*32
      CLENGT(2) = NELBLK*8
      CLENGT(3) = NUQPRP*8
C     ...Compute MORMEMR and check for ZERO length (must be at least=1)
      DO 10 I = BRDAIDX, ERDAIDX, 1
         IF(RLENGT(I).EQ.0)THEN
            RLENGT(I) = 1
         ELSEIF(RLENGT(I).LT.0)THEN
            IERR = IERR + 1
            WRITE(NOUTFL,'(A,I3,A)')' ***RLENGT(',I,') < 0***'
            WRITE(NOUTFL,'(A,I3,2A)')' RNAMES(',I,')=',RNAMES(I)
         ENDIF
         MORMEMR = MORMEMR + RLENGT(I)
   10 CONTINUE
      IF(DEBUG)WRITE(NOUTFL,'(A)')' $$$$$$$$$$$$$$$$$$$$$$$$$$$$$'//
     $'$$ ADDMEM $$$$$$$$$$$$$$$$$$$$$$$$$$$$'
      IF(IERR.GT.0)THEN
         CALL QAABORT(
     $   '(ADDMEM) - REAL array size errors detected ***')
      ELSEIF(MORMEMR.LT.0)THEN
         WRITE(NOUTFL,*)' **** MORMEMR < 0 in ADDMEM ****'
         DO 15 I=BRDAIDX,ERDAIDX
            WRITE(NOUTFL,'(1X,2A,I2,A,I9)')
     $      RNAMES(I),' RLENGT(',I,')=',RLENGT(I)
   15    CONTINUE
         CALL QAABORT(
     $   '(ADDMEM) - REAL array memory errors detected ***')
      ENDIF
C
C     ...Compute MORMEMC and check for ZERO length (must be at least=1)
      DO 20 I = BCDAIDX, ECDAIDX, 1
         IF(CLENGT(I).EQ.0)THEN
            CLENGT(I) = 1
         ELSEIF(CLENGT(I).LT.0)THEN
            IERR = IERR + 1
            WRITE(NOUTFL,'(A,I3,A)')' ***CLENGT(',I,') < 0***'
            WRITE(NOUTFL,'(A,I3,2A)')' CNAMES(',I,')=',CNAMES(I)
         ENDIF
         MORMEMC = MORMEMC + CLENGT(I)
   20 CONTINUE
      IF(IERR.GT.0)THEN
         CALL QAABORT(
     $   '(ADDMEM) - CHARACTER array size errors detected ***')
      ELSEIF(MORMEMC.LT.0)THEN
         WRITE(NOUTFL,*)' **** MORMEMC < 0 in ADDMEM ****'
         DO 25 I=BCDAIDX,ECDAIDX
            WRITE(NOUTFL,'(1X,2A,I2,A,I9)')
     $      CNAMES(I),' CLENGT(',I,')=',CLENGT(I)
   25    CONTINUE
         CALL QAABORT(
     $   '(ADDMEM) - CHARACTER array memory errors detected ***')
      ENDIF
      RETURN
C
```

```
449   C*********************************************************************
450   C
451         ENTRY ADDMEM1( MORMEMR, MORMEMC )
452   C
453   C*********************************************************************
454   C
455   C     THIS SECTION COMPUTES THE MEMORY REQUIRED FOR
456   C     POSTINFIL including net infitration, AND OTHER RELATED DYNAMIC ARRAYS:
457   C
458   C     (CALLED BY PROCIN)
459   C
460         IERR       = 0
461         MORMEMC    = 0
462         MORMEMR    = 0
463   C     ...Set beginning and ending dynamic array indices
464         BRDAIDX    = 5
465         ERDAIDX    = 6
466         BCDAIDX    = 4
467         ECDAIDX    = 5
468   C     ...Set dynamic REAL array lengths
469   C     ...()  AVE_INFIL, NUMCELLS, RUNOFF previously stored and newly stored
470         RLENGT(5) = NVARGL*24
471         RLENGT(6) = NVARGL2*4
472   C     ...Set dynamic CHARACTER*1 array lengths
473         CLENGT(4) = 8*NVARGL*8
474         CLENGT(5) = 8*NVARGL2
475   C*      CLENGT(6) = 8*NVARGL2
476   C     ...Check for ZERO length (must be at least=1)
477         DO 200 I = BRDAIDX, ERDAIDX, 1
478            IF(RLENGT(I).EQ.0)THEN
479               RLENGT(I) = 1
480            ELSEIF(RLENGT(I).LT.0)THEN
481               IERR = IERR + 1
482               WRITE(NOUTFL,'(A,I3,A)')' ***RLENGT(',I,') < 0***'
483               WRITE(NOUTFL,'(A,I3,2A)')' RNAMES(',I,')=',RNAMES(I)
484            ENDIF
485            MORMEMR = MORMEMR + RLENGT(I)
486    200  CONTINUE
487         IF(DEBUG)WRITE(NOUTFL,'(A)')' $$$$$$$$$$$$$$$$$$$$$$$$$$$$'//
488        $'$$ ADDMEM2 $$$$$$$$$$$$$$$$$$$$$$$$$$$$'
489         IF(IERR.GT.0)THEN
490            CALL QAABORT('(ADDMEM2) - REAL array size errors detected ***')
491         ELSEIF(MORMEMR.LT.0)THEN
492            WRITE(NOUTFL,*)' **** MORMEMR < 0 in ADDMEM2 ****'
493            DO 210 I=BRDAIDX,ERDAIDX
494               WRITE(NOUTFL,'(1X,2A,I2,A,I9)')
495        $      RNAMES(I),' RLENGT(',I,')=',RLENGT(I)
496    210     CONTINUE
497            CALL QAABORT('(ADDMEM2) - REAL memory errors detected ***')
498         ENDIF
499
500   C     ...Check for ZERO length (must be at least=1)
501         DO 220 I = BCDAIDX, ECDAIDX, 1
502            IF(CLENGT(I).EQ.0)THEN
503               CLENGT(I) = 1
504            ELSEIF(CLENGT(I).LT.0)THEN
505               IERR = IERR + 1
```

```
506              WRITE(NOUTFL,'(A,I3,A)')' ***CLENGT(',I,') < 0***'
507              WRITE(NOUTFL,'(A,I3,2A)')' CNAMES(',I,')=',CNAMES(I)
508           ENDIF
509           MORMEMC = MORMEMC + CLENGT(I)
510    220  CONTINUE
511        IF(IERR.GT.0)THEN
512           CALL QAABORT(
513       $   '(ADDMEM2) - CHARACTER array size errors detected ***')
514        ELSEIF(MORMEMC.LT.0)THEN
515           WRITE(NOUTFL,*)' **** MORMEMC < 0 in ADDMEM2 ****'
516           DO 230 I=BCDAIDX,ECDAIDX
517              WRITE(NOUTFL,'(1X,2A,I2,A,I9)')
518       $      CNAMES(I),' CLENGT(',I,')=',CLENGT(I)
519    230    CONTINUE
520           CALL QAABORT(
521       $   '(ADDMEM2) - CHARACTER array memory errors detected ***')
522        ENDIF
523
524        RETURN
525  C-----------------------------------------------------------------------
526  C**** END OF SUBROUTINE ADDMEM ****
527  C-----------------------------------------------------------------------
528        END
529  *DECK BLOCK
530        BLOCK DATA BLOCK
531  C***********************************************************************
532  C
533  C     PURPOSE:    Initializes variables that are in COMMON BLOCKS
534  C
535  C     AUTHOR:     Ron McCurley, [NMERI]
536  C
537  C     UPDATED:    17-JAN-1999
538  C                 19-AUG-1999 RDM: modified for adding INFIL summary file
539  C
540  C     ARGUMENTS:
541  C     ENTRY/
542  C       none
543  C
544  C     LOCAL/
545  C       none
546  C
547  C     EXIT/
548  C      --through common blocks:
549  C       /CONSTA/ (from FORTRAN statement: INCLUDE 'CONSTA.INC')
550  C        AVOGAD = Avogadro's number [DOUBLE PRECISION]
551  C        ACCNST = Activity Conversion CoNStanT [DOUBLE PRECISION]
552  C                 A radioisotope's activity conversion is equal
553  C                 to ACCNST/(half_life*Atomic_weight)
554  C        PI     = Circumference/Radius [DOUBLE PRECISION]
555  C
556  C       /DYNAME/ (from FORTRAN statement: INCLUDE 'DYNAME.INC')
557  C        CNAMES = Names of CHARACTER dynamic arrays (CHARACTER*8)
558  C        DTYPES = Types of REAL dynamic arrays (CHARACTER*8)
559  C        RNAMES = Names of REAL data dynamic arrays (CHARACTER*8)
560  C
561  C       /FILEUN/ (from FORTRAN statement: INCLUDE 'FILEUN.INC')
562  C        IDB    = Device no. of input CAMDAT file
```

```
563   C         IDBSCR = Device no. of CAMDAT scratch file
564   C         OUTPUTF = Device no. of input text file
565   C         ISCRAT = Device no. of SCRATCH file
566   C         NOUTFL = Device no. of output diagnostics/debug file
567   C         ODB    = Device no. of output CAMDAT file
568   C
569   C        /INDEX/ (from FORTRAN statement: INCLUDE 'INDEX.INC')
570   C         LASCIDX= Last index of CHARACTER*1 dynamic arrays [INTEGER]
571   C         LASRIDX= Last index of REAL dynamic arrays [INTEGER]
572   C
573   C        /MACHDP/ (from FORTRAN statement: INCLUDE 'MACHDP.INC')
574   C         XLIMDP = Machine limit for zero (REAL*8)
575   C         XMINDP = Machine limit lower exponent value (REAL*8)
576   C         XMAXDP = Machine limit upper exponent value (REAL*8)
577   C         VALMAXD= Largest DOUBLE PRECISION real value allowed (REAL*8)
578   C
579   C        /MACHINE/ (from FORTRAN statement: INCLUDE 'MACHINE.INC')
580   C         XLIM   = Machine limit for zero (REAL)
581   C         XMIN   = Machine limit lower exponent value (REAL)
582   C         XMAX   = Machine limit upper exponent value (REAL)
583   C         VALMAX = Largest SINGLE PRECISION real value allowed (REAL)
584   C
585   C        /MISCDAT/ (from FORTRAN statement: INCLUDE 'MISCDAT.INC')
586   C         NSOLNS  = No. of solution times [INTEGER]
587   C
588   C        /QA/     (from FORTRAN statement: INCLUDE 'QA.INC')
589   C         PRGNAM = Program name    (CHARACTER*8)
590   C         PRGVER = "       version (CHARACTER*8)
591   C         PRGUPD = "       update  (CHARACTER*8)
592   C         PRGATH = "       author  (CHARACTER*80)
593   C         PRGSPN = "       sponsor (CHARACTER*80)
594   C
595   C        /TOOLS/  (from FORTRAN statement: INCLUDE 'TOOLS.INC')
596   C         DEBUG  = Logical for debugging dynamic memory allocation
597   C
598   C*********************************************************************
599   C234567
600         IMPLICIT NONE
601         INCLUDE 'CONSTA.INC'
602         INCLUDE 'DYNAME.INC'
603         INCLUDE 'FILEUN.INC'
604         INCLUDE 'INDEX.INC'
605         INCLUDE 'MACHDP.INC'
606         INCLUDE 'MACHINE.INC'
607         INCLUDE 'MISCDAT.INC'
608         INCLUDE 'PARAMS.INC'
609         INCLUDE 'QA.INC'
610         INCLUDE 'TOOLS.INC'
611   C
612   C<><><><><><><><><><><><>
613   C...Begin Procedures...
614   C<><><><><><><><><><><><>
615   C
616   C     ##############################
617   C     ...Define conversion constants
618   C     ##############################
619   C     ...Common mathematical constants (COMMON /CONSTA/)
```

```
620   C     ...Source: "Preliminary Performance Assessment for the Waste
621   C                 Isolation Pilot Plant, December 1992 "
622   C                 Volume 3: Model Parameters., Table 15,
623   C                 p. Conversion Tables - 10.
624         DATA ACCNST/1.128D+13/
625         DATA AVOGAD/6.022D+23/
626
627   C     ...Dynamic array names (COMMON /DYNAME/)
628         DATA DTYPES/
629        $'INTEGER','INTEGER','LOGICAL','REAL',
630        $'INTEGER','REAL'/
631         DATA RNAMES/
632        $'IDEBLK', 'NUMPRP', 'IASPRP', 'XMATPR',
633        $'NUMCELL','AVEINF'/
634         DATA CNAMES/
635        $'QAINFO','NAMELB','NMATPR','SAVNAM','NAMNUM','NAMINF'/
636
637   C     ...File unit variables (COMMON /FILEUN/)
638         DATA IDB/30/IDBSCR/2/ICNTRLFIL/8/INFLOUT1/9/INFLOUT2/10/ISCRAT/16/
639        $     ODB/31/
640
641   C     ...Dynamic array variables (COMMON /INDEX/)
642         DATA LASCIDX/6/LASRIDX/6/
643
644   C     ...Machine limit variables: DOUBLE PRECISION (COMMON /MACHDP/)
645         DATA XMINDP, XMAXDP, VALMAXD / -709.19D0, 706.89D0, 1.0D+307 /
646         DATA XLIMDP/1.0D-15/
647
648   C     ...Machine limit variables (COMMON /MACHINE/)
649         DATA XMIN, XMAX, VALMAX / -88.75, 88.00, 1.0E+38 /
650         DATA XLIM/1.0E-07/
651
652   C     ...Number of solution times (COMMON /MISCDAT/)
653         DATA NSOLNS/1/
654
655   C     ...QA variables (COMMON /QA/)
656         DATA PRGNAM/'POSTINFIL'/,PRGVER/'A2.70'/,PRGUPD/'11/21/99'/,
657        $     PRGATH/'Ron McCurley, [NMERI]'/,
658        $     PRGSPN/'Ron McCurley, [NMERI]'/
659
660   C     ...Debugging variables (COMMON /TOOLS/)
661         DATA DEBUG/.FALSE./
662         DATA GLOBAL/.TRUE./
663   C
664   C-----------------------------------------------------------------------
665   C**** END OF BLOCKDATA BLOCK ****
666   C-----------------------------------------------------------------------
667   C
668         END
669   *DECK DYNMEM
670         SUBROUTINE DYNMEM( MORMEMR, MEMORY, Q )
671   C***********************************************************************
672   C***                                                                 ***
673   C***          applies D_Y_N_amic  M_E_M_eory  allocation             ***
674   C***                  - - -       - - -                              ***
675   C***********************************************************************
676   C
```

```
677   C   PURPOSE:      Allocates MEMORY DYNAMICally
678   C
679   C   AUTHOR:       Jonathan S. Rath
680   C
681   C   UPDATED:      19-DEC-1991  --Adapted for REAL dynamic base array
682   C                 01-FEB-1990  --Fixed record lengths <= 72 chars.
683   C                 14-DEC-1989  --Increased INTEGER formats
684   C                 13-JUN-1989  --First Ed.
685   C
686   C   CALLED BY:    PROCDB
687   C                 PROCIN
688   C                 PROCQA
689   C
690   C   ENTRY POINT: DYNMEM2, DYNMEM3
691   C
692   C   CALLS:        MEMADJ
693   C                 MEMADJ2
694   C                 MEMADJ3
695   C
696   C   ARGUMENTS:
697   C   ENTRY/
698   C     --common block
699   C      /FILEUN/ ($include 'FILEUN.INC')
700   C       NOUTFL = File unit of diagnostics/debug output file
701   C
702   C     --through subroutine call
703   C       MORMEMR = Add'l words needed for REAL core allocation (Q)
704   C       Q       = Dynamic memory REAL base array
705   C
706   C   LOCAL/
707   C       MEMADD = Actual number of words added to core (REAL-Q)
708   C       MEMOLD = Previous total no. of words allocated to core (Q)
709   C
710   C   EXIT/
711   C     --through subroutine call
712   C       MEMORY = Total number of words allocated to core (Q-REAL)
713   C       Q      = Dynamic memory REAL base array
714   C
715   C*********************************************************************
716   C234567
717         IMPLICIT NONE
718         INCLUDE 'FILEUN.INC'
719         INTEGER MEMADD, MEMOLD, MEMORY, MORMEMR
720         REAL Q(*)
721   C
722   C*********************************************************************
723   C
724   C     PSEUDO ENTRY DYNMEM( MORMEMR, MEMORY, Q )
725   C
726   C*********************************************************************
727   C
728         MEMOLD = MEMORY
729         CALL MEMADJ( MORMEMR, MEMORY, Q )
730         MEMADD = MEMORY - MEMOLD
731         WRITE(NOUTFL,900) MEMADD, 'DYNMEM', MEMORY
732         RETURN
733   C
```

```
C*********************************************************************
C
      ENTRY DYNMEM2( MORMEMR, MEMORY, Q )
C
C*********************************************************************
C
C     THIS SECTION ADJUSTS MEMORY ALLOCATION BY LENGTHENING
C     (OR SHORTENING) CERTAIN DYNAMIC ARRAYS
C
      MEMOLD = MEMORY
      CALL MEMADJ2( MORMEMR, MEMORY, Q )
      MEMADD = MEMORY - MEMOLD
      WRITE(NOUTFL,900) MEMADD, 'DYNMEM2', MEMORY
      IF(MORMEMR.GT.MEMADD)WRITE(NOUTFL,910)MORMEMR,MEMADD,MEMOLD,MEMORY
      RETURN
C
C*********************************************************************
C
      ENTRY DYNMEM3( MEMORY, Q )
C
C*********************************************************************
C
C     THIS SECTION IS USED TO CONSOLIDATE MEMORY AND/OR FOR
C     ADJUSTING MEMORY ALLOCATION DUE TO "PSEUDO-DELETING"
C     DYNAMIC ARRAYS.  "PSEUDO-DELETING" MEANS SETTING THE UNUSED
C     DYNAMIC ARRAYS TO LENGTH 1.
C
      MEMOLD = MEMORY
      CALL MEMADJ3( MEMORY, Q )
      MEMADD = MEMORY - MEMOLD
      WRITE(NOUTFL,900) MEMADD, 'DYNMEM3', MEMORY
      RETURN
C
C---------------  F O R M A T   S T A T E M E N T S  -----------------
C
 900  FORMAT(
     +//5X,'-------------------------------------------------------',
     + /5X,'       ',I10,' more words allocated by ',A,
     + /5X,'           Total allocated is ',I10,
     + /5X,'-------------------------------------------------------',//)
 910  FORMAT( //,'***** Possible Error in  D y n a m i c  Memory',
     +          ' Allocation (Q) *****',
     +  /,5X,' Min. Additional words REQ"D     = ',I10,' (MORMEMR)',
     +  /,5X,' Additional words added          = ',I10,' (MEMADD)',
     +  /,5X,' Previous total word allocation  = ',I10,' (MEMOLD)',
     +  /,5X,' Current total word allocation   = ',I10,' (MEMORY)',
     +       //,'*********************************************',
     +          '******************' )
C
C---------------------------------------------------------------------
C**** END OF SUBOUTINE DYNMEM ****
C---------------------------------------------------------------------
C
      END
*DECK DYNMEMC
      SUBROUTINE DYNMEMC( MORMEMC, MEMORY, C )
C*********************************************************************
```

```
C***                                                                  ****
C***  applies D_Y_N_amic  M_E_M_ory  (C_haracter type) allocation     ****
C***          - - -       - - -       -                               ****
C*********************************************************************
C
C   PURPOSE:     Allocates MEMORY DYNAMICally to CHARACTER base array
C
C   AUTHOR:      Jonathan S. Rath
C
C   UPDATED:     02-DEC-1991  --First Ed.
C
C   CALLED BY:   PROCDB
C                PROCIN
C                PROCQA
C
C   ENTRY POINT: DYNMEMC2, DYNMEMC3
C
C   CALLS:       MEMADC
C                MEMADJC2
C                MEMADJC3
C   ARGUMENTS:
C   ENTRY/
C     --common block
C      /FILEUN/ ($include 'FILEUN.INC')
C       NOUTFL = File unit of diagnostics/debug output file
C
C     --through subroutine call
C       MORMEMC = Add'l words needed for CHAR core allocation (C)
C       C       = Dynamic memory CHARACTER*1 base array
C
C   LOCAL/
C       MEMADD = Actual number of words added to core (CHAR-C)
C       MEMOLD = Previous total no. of words allocated to core (C)
C
C   EXIT/
C     --through subroutine call
C       MEMORY = Total number of words allocated to core (C-CHAR)
C       C      = Dynamic memory CHARACTER*1 base array
C
C*********************************************************************
C234567
      IMPLICIT NONE
      INCLUDE 'FILEUN.INC'
      INTEGER MEMADD, MEMOLD, MEMORY, MORMEMC
      CHARACTER*(1) C(*)
C
C*********************************************************************
C
C     PSEUDO ENTRY DYNMEMC( MORMEMC, MEMORY, C )
C
C*********************************************************************
C
      MEMOLD = MEMORY
      CALL MEMADJC( MORMEMC, MEMORY, C )
      MEMADD = MEMORY - MEMOLD
      WRITE(NOUTFL,900) MEMADD, 'DYNMEMC', MEMORY
      RETURN
```

```
C
C*********************************************************************
C
      ENTRY DYNMEMC2( MORMEMC, MEMORY, C )
C
C*********************************************************************
C
C     THIS SECTION ADJUSTS MEMORY ALLOCATION BY LENGTHENING
C     (OR SHORTENING) CERTAIN DYNAMIC ARRAYS
C
      MEMOLD = MEMORY
      CALL MEMADJC2( MORMEMC, MEMORY, C )
      MEMADD = MEMORY - MEMOLD
      WRITE(NOUTFL,900) MEMADD, 'DYNMEMC2', MEMORY
      IF(MORMEMC.GT.MEMADD)WRITE(NOUTFL,910)MORMEMC,MEMADD,MEMOLD,MEMORY
      RETURN
C
C*********************************************************************
C
      ENTRY DYNMEMC3( MEMORY, C )
C
C*********************************************************************
C
C     THIS SECTION CAN BE USED FOR MEMORY CONSOLIDATION AND/OR FOR
C     ADJUSTING MEMORY ALLOCATION DUE TO "PSEUDO-DELETING"
C     DYNAMIC ARRAYS.  "PSEUDO-DELETING" MEANS SETTING THE UNUSED
C     DYNAMIC ARRAYS TO LENGTH 1.
C
      MEMOLD = MEMORY
      CALL MEMADJC3( MEMORY, C )
      MEMADD = MEMORY - MEMOLD
      WRITE(NOUTFL,900) MEMADD, 'DYNMEMC3', MEMORY
      RETURN
C
C---------------  F O R M A T   S T A T E M E N T S  -----------------
C
 900  FORMAT(
     +//5X,'-------------------------------------------------------',
     + /5X,' ',I10,' more CHAR*1 words allocated by ',A,
     + /5X,'           Total allocated is ',I10,
     + /5X,'-------------------------------------------------------',//)
 910  FORMAT( //,'***** Possible Error in  D y n a m i c  Memory',
     +          ' Allocation (C) *****',
     +  /,5X,' Min. Additional words REQ"D     = ',I10,' (MORMEMC)',
     +  /,5X,' Additional words added           = ',I10,' (MEMADD)',
     +  /,5X,' Previous total word allocation  = ',I10,' (MEMOLD)',
     +  /,5X,' Current total word allocation   = ',I10,' (MEMORY)',
     +       //,'*********************************************',
     +          '*******************' )
C
C-----------------------------------------------------------------------
C**** END OF SUBOUTINE DYNMEMC ****
C-----------------------------------------------------------------------
C
      END
*DECK MEMADJ
      SUBROUTINE MEMADJ( MORMEM, MNUSED, Q )
```

```
905   C*********************************************************************
906   C***                                                               ****
907   C***                A_D_J_usts  dynamic  M_E_M_ory                 ****
908   C***               - - -                - - -                      ****
909   C*********************************************************************
910   C
911   C  PURPOSE:      Adjusts the amount of dynamic memory allocated
912   C                the to REAL base array, Q.
913   C
914   C  AUTHOR:       Jonathan S. Rath
915   C
916   C  UPDATED:      13-JUN-1994  --JSR: First Edition
917   C
918   C  CALLED BY:    DYNMEM
919   C                DYNMEM2 (entry point)
920   C                DYNMEM3 (entry point)
921   C
922   C  ENTRY POINT: MEMADJ2
923   C                MEMADJ3
924   C
925   C  CALLS:        (MEM_MGR):  MDGET        (V_EXT_LIB): EXEMEM
926   C                            MDRSRV
927   C                            MDLONG
928   C                            MDLIST
929   C                            MDSTAT
930   C                            MDEROR
931   C                            MDDEBG
932   C                            MDDEL
933   C                            MDCOMP
934   C                            MDGIV
935   C
936   C  ARGUMENTS:
937   C  ENTRY/
938   C    --common blocks
939   C      /FILEUN/ (from FORTRAN statement: 'FILEUN.INC')
940   C       NOUTFL = File unit of diagnostics/debug output file
941   C
942   C      /LENGTH/ (from FORTRAN statement: 'LENGTH.INC')
943   C       RLENGT = Dynamic REAL arrays LENGTHS array
944   C
945   C      /DYNAME/ (from FORTRAN statement: 'DYNAME.INC')
946   C       RNAMES = Dynamic REAL arrays NAMES array
947   C
948   C      /INDEX/ (from FORTRAN statement: 'INDEX.INC')
949   C       BRDAIDX = Beg. Real Dynamic Array InDeX for memory allocation
950   C       ERDAIDX = End. Real "
951   C
952   C      /TOOLS/ (from FORTRAN statement: 'TOOLS.INC')
953   C       DEBUG  = Logical debugging tool
954   C
955   C     --through subroutine call
956   C       MORMEM = Add'l no. of words needed for core allocation (REAL-Q)
957   C       Q      = Dynamic Memory base REAL array
958   C
959   C  LOCAL/
960   C       MNERRS =  Number of Memory manager ERRorS during execution
961   C
```

```
962   C  EXIT/
963   C     --common blocks
964   C      /LOCAT/ (from FORTRAN statement: 'LOCAT.INC')
965   C       LRPTR  = Pointer array used for REAL dynamic array (Q)
966   C
967   C     --through subroutine call
968   C       MNUSED = Total words allocated to REAL dynamic array
969   C       Q      = Dynamic Memory base REAL array
970   C
971   C*********************************************************************
972   C234567
973         IMPLICIT NONE
974         INCLUDE 'FILEUN.INC'
975         INCLUDE 'LOCAT.INC'
976         INCLUDE 'DYNAME.INC'
977         INCLUDE 'LENGTH.INC'
978         INCLUDE 'INDEX.INC'
979         INCLUDE 'TOOLS.INC'
980         INTEGER I, I1, I2, MORMEM, MNERRS, MNUSED
981         REAL Q(*)
982   C
983   C*********************************************************************
984   C
985   C     PSEUDO ENTRY MEMADJ( MORMEM, MNUSED, Q )
986   C
987   C*********************************************************************
988   C
989   C     THIS SECTION SETS DYNAMIC ARRAY LENGTHS FOR FIRST TIME
990   C
991   C     ...Check for bad REAL dynamic array indices
992         IF(BRDAIDX.LT.1.OR.ERDAIDX.GT.LASRIDX.OR.
993        +   ERDAIDX.LT.1.OR.BRDAIDX.GT.LASRIDX    )THEN
994            WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
995            WRITE(NOUTFL,*)'     BRDAIDX = ',BRDAIDX
996            WRITE(NOUTFL,*)'     ERDAIDX = ',ERDAIDX
997            WRITE(NOUTFL,*)'     LASRIDX = ',LASRIDX
998            CALL QAABORT('(MEMADJ) - Invalid REAL dynamic array '//
999        +              'indices detected ***')
1000        ENDIF
1001        CALL MDGET(MORMEM)
1002        CALL MDFILL(1.0)
1003        DO 100 I = BRDAIDX, ERDAIDX, 1
1004           CALL MDRSRV(RNAMES(I),I1,RLENGT(I))
1005           LRPTR(I) = I1
1006   100  CONTINUE
1007  C     ...Check for dynamic memory allocation errors
1008        IF(DEBUG)CALL MDDEBG(NOUTFL)
1009        CALL MDSTAT(MNERRS,MNUSED)
1010        IF(MNERRS.NE.0.OR.DEBUG)CALL MDLIST(NOUTFL)
1011        IF(MNERRS.NE.0.        )THEN
1012           CALL MDEROR(NOUTFL)
1013           CALL QAABORT('(MEMADJ) - REAL DYNAMIC MEMORY '//
1014       +              'ALLOCATION ERRORS ***')
1015        ENDIF
1016        RETURN
1017  C
1018  C*********************************************************************
```

```
1019  C
1020        ENTRY MEMADJ2( MORMEM, MNUSED, Q )
1021  C
1022  C*********************************************************************
1023  C
1024  C     THIS SECTION LENGTHENS OR SHORTENS DYNAMIC ARRAYS
1025  C
1026  C     ...Check for bad REAL dynamic array indices
1027        IF(BRDAIDX.LT.1.OR.ERDAIDX.GT.LASRIDX.OR.
1028       +   ERDAIDX.LT.1.OR.BRDAIDX.GT.LASRIDX    )THEN
1029           WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
1030           WRITE(NOUTFL,*)'     BRDAIDX = ',BRDAIDX
1031           WRITE(NOUTFL,*)'     ERDAIDX = ',ERDAIDX
1032           WRITE(NOUTFL,*)'     LASRIDX = ',LASRIDX
1033           CALL QAABORT('(MEMADJ2) - Invalid REAL dynamic array '//
1034       +              'indices detected ***')
1035        ENDIF
1036        DO 200 I = BRDAIDX, ERDAIDX, 1
1037           CALL MDLONG(RNAMES(I),I1,RLENGT(I))
1038           LRPTR(I) = I1
1039   200  CONTINUE
1040  C     ...Check for dynamic memory allocation errors
1041        IF(DEBUG)CALL MDDEBG(NOUTFL)
1042        CALL MDSTAT(MNERRS,MNUSED)
1043        IF(MNERRS.NE.0.OR.DEBUG)CALL MDLIST(NOUTFL)
1044        IF(MNERRS.NE.0.        )THEN
1045           CALL MDEROR(NOUTFL)
1046           CALL QAABORT('(MEMADJ2) - REAL '//
1047       $                'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1048        ENDIF
1049        RETURN
1050  C
1051  C*********************************************************************
1052  C
1053        ENTRY MEMADJ3( MNUSED, Q )
1054  C
1055  C*********************************************************************
1056  C
1057  C     THIS SECTION CONSOLIDATES FRAGMENTED MEMORY, RETURNS UNUSED
1058  C     STORAGE TO THE OPERATING SYSTEM, AND THEN RECALCULATES ALL
1059  C     DYNAMIC ARRAY POINTERS.
1060  C
1061  C
1062  C     ...Consolidate the fragmented memeory
1063        CALL MDCOMP
1064  C     ...Return unused storage back to the operating system
1065        CALL MDGIVE
1066  C     ...Recalculate ALL pointers of the Q-array
1067        DO 300 I = BRDAIDX, ERDAIDX, 1
1068           I2 = 0
1069           CALL MDFIND(RNAMES(I),I1,I2)
1070           LRPTR(I)  = I1
1071           RLENGT(I) = I2
1072   300  CONTINUE
1073  C     ...Check for dynamic memeory allocation errors
1074        IF(DEBUG)CALL MDDEBG(NOUTFL)
1075        CALL MDSTAT(MNERRS,MNUSED)
```

```
1076          IF(MNERRS.NE.0.OR.DEBUG)CALL MDLIST(NOUTFL)
1077          IF(MNERRS.NE.0.         )THEN
1078             CALL MDEROR(NOUTFL)
1079             CALL QAABORT('(MEMADJ3) - REAL '//
1080         $                 'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1081          ENDIF
1082          RETURN
1083   C-----------------------------------------------------------------------
1084   C**** END OF SUBROUTINE MEMADJ ****
1085   C-----------------------------------------------------------------------
1086          END
1087   *DECK MEMADJC
1088          SUBROUTINE MEMADJC( MORMEMC, MNUSED, C )
1089   C***********************************************************************
1090   C***                                                                 ****
1091   C***       A_D_J_usts  dynamic  M_E_M_ory  (C_haracter type)     .   ****
1092   C***       - - -                - - -      -                         ****
1093   C***********************************************************************
1094   C
1095   C  PURPOSE:      Adjusts the amount of dynamic memory allocated to
1096   C                the CHARACTER*1 base array, C.
1097   C
1098   C  AUTHOR:       Jonathan S. Rath
1099   C
1100   C  UPDATED:      13-JUN-1994  --JSR: First Edition
1101   C
1102   C  CALLED BY:    DYNMEMC
1103   C                DYNMEMC2 (entry point)
1104   C                DYNMEMC3 (entry point)
1105   C
1106   C  ENTRY POINT: MEMADJC2
1107   C                MEMADJC3
1108   C
1109   C  CALLS:        (MEM_MGR):  MCGET        (V_EXT_LIB): EXEMEM
1110   C                            MCRSRV
1111   C                            MCLONG
1112   C                            MCLIST
1113   C                            MCSTAT
1114   C                            MCEROR
1115   C                            MCFIL
1116   C                            MCDEBG
1117   C                            MCCOMP
1118   C                            MCGIV
1119   C
1120   C  ARGUMENTS:
1121   C  ENTRY/
1122   C    --common blocks
1123   C      /FILEUN/ (from FORTRAN statement: 'FILEUN.INC')
1124   C       NOUTFL = File unit of diagnostics/debug output file
1125   C
1126   C      /LENGTH/ (from FORTRAN statement: 'LENGTH.INC')
1127   C       CLENGT = Dynamic CHAR arrays LENGTHS array
1128   C
1129   C      /DYNAME/ (from FORTRAN statement: 'DYNAME.INC')
1130   C       CNAMES = Dynamic CHAR arrays NAMES array
1131   C
1132   C      /INDEX/ (from FORTRAN statement: 'INDEX.INC')
```

```
1133  C        BCDAIDX = Beg. Char Dynamic Array InDeX for memory allocation
1134  C        ECDAIDX = End. Char "
1135  C
1136  C       /TOOLS/ (from FORTRAN statement: 'TOOLS.INC')
1137  C        DEBUG  = Logical debugging tool
1138  C
1139  C      --through subroutine call
1140  C        MORMEMC = Add'l no. of words needed for core allocation (CHAR-C)
1141  C        C      = Dynamic memory CHARACTER*1 base array
1142  C
1143  C      --through ENTRY POINT
1144  C        NAME   = Name of array to be deleted
1145  C
1146  C  LOCAL/
1147  C        MNERRS =  Number of Memory manager ERRorS during execution
1148  C
1149  C  EXIT/
1150  C      --common blocks
1151  C       /LOCAT/ (from FORTRAN statement: 'LOCAT.INC')
1152  C        LCPTR  = Pointer array used for CHAR dynamic array (C)
1153  C
1154  C      --through subroutine call
1155  C        MNUSED = Total words allocated to CHARACTER dynamic array
1156  C        C      = Dynamic Memory CHARACTER*1 base array
1157  C
1158  C*********************************************************************
1159  C234567
1160        IMPLICIT NONE
1161        INCLUDE 'FILEUN.INC'
1162        INCLUDE 'LOCAT.INC'
1163        INCLUDE 'DYNAME.INC'
1164        INCLUDE 'LENGTH.INC'
1165        INCLUDE 'INDEX.INC'
1166        INCLUDE 'TOOLS.INC'
1167        INTEGER I, I1, I2, MORMEMC, MNERRS, MNUSED
1168        CHARACTER*(1) C(*)
1169  C
1170  C*********************************************************************
1171  C
1172  C     PSEUDO ENTRY MEMADJC( MORMEMC, MNUSED, C )
1173  C
1174  C*********************************************************************
1175  C
1176  C     THIS SECTION SETS DYNAMIC ARRAY LENGTHS FOR FIRST TIME
1177  C
1178  C     ...Check for bad CHARACTER dynamic array indices
1179        IF(BCDAIDX.LT.1.OR.ECDAIDX.GT.LASCIDX.OR.
1180     +    ECDAIDX.LT.1.OR.BCDAIDX.GT.LASCIDX    )THEN
1181           WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
1182           WRITE(NOUTFL,*)'     BCDAIDX = ',BCDAIDX
1183           WRITE(NOUTFL,*)'     ECADIDX = ',ECDAIDX
1184           WRITE(NOUTFL,*)'     LASCIDX = ',LASCIDX
1185           CALL QAABORT('(MEMADJ) - Invalid CHARACTER dynamic array '//
1186     +                 'indices detected ***')
1187        ENDIF
1188        CALL MCGET(MORMEMC)
1189        CALL MCFILL(' ')
```

```
1190          DO 100 I = BCDAIDX, ECDAIDX, 1
1191             CALL MCRSRV(CNAMES(I),I1,CLENGT(I))
1192             LCPTR(I) = I1
1193    100   CONTINUE
1194  C       ...Check for dynamic memory allocation errors
1195          IF(DEBUG)CALL MCDEBG(NOUTFL)
1196          CALL MCSTAT(MNERRS,MNUSED)
1197          IF(MNERRS.NE.0.OR.DEBUG)CALL MCLIST(NOUTFL)
1198          IF(MNERRS.NE.0.         )THEN
1199             CALL MCEROR(NOUTFL)
1200             CALL QAABORT('(MEMADJC) - CHARACTER DYNAMIC MEMORY '//
1201         +                'ALLOCATION ERRORS ***')
1202          ENDIF
1203          RETURN
1204  C
1205  C*********************************************************************
1206  C
1207          ENTRY MEMADJC2( MORMEMC, MNUSED, C )
1208  C
1209  C*********************************************************************
1210  C
1211  C       THIS SECTION LENGTHENS OR SHORTENS DYNAMIC ARRAYS
1212  C
1213  C       ...Check for bad CHARACTER dynamic array indices
1214          IF(BCDAIDX.LT.1 .OR. ECDAIDX.GT.LASCIDX .OR.
1215         +   ECDAIDX.LT.1 .OR. BCDAIDX.GT.LASCIDX    )THEN
1216             WRITE(NOUTFL,*)' *** Bad dynamic array indices ***'
1217             WRITE(NOUTFL,*)'     BCDAIDX = ',BCDAIDX
1218             WRITE(NOUTFL,*)'     ECADIDX = ',ECDAIDX
1219             WRITE(NOUTFL,*)'     LASCIDX = ',LASCIDX
1220             CALL QAABORT('(MEMADJC2) - Invalid CHARACTER dynamic array '//
1221         +                'indices detected ***')
1222          ENDIF
1223          DO 200 I = BCDAIDX, ECDAIDX, 1
1224             CALL MCLONG(CNAMES(I),I1,CLENGT(I))
1225             LCPTR(I) = I1
1226    200   CONTINUE
1227          CALL MCFILL(' ')
1228  C       ...Check for dynamic memory allocation errors
1229          IF(DEBUG)CALL MCDEBG(NOUTFL)
1230          CALL MCSTAT(MNERRS,MNUSED)
1231          IF(MNERRS.NE.0.OR.DEBUG)CALL MCLIST(NOUTFL)
1232          IF(MNERRS.NE.0.         )THEN
1233             CALL MCEROR(NOUTFL)
1234             CALL QAABORT('(MEMADJC2) - CHARACTER '//
1235         $                'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1236          ENDIF
1237          RETURN
1238  C
1239  C*********************************************************************
1240  C
1241          ENTRY MEMADJC3( MNUSED, C )
1242  C
1243  C*********************************************************************
1244  C
1245  C       THIS SECTION CONSOLIDATES FRAGMENTED MEMORY, RETURNS UNUSED
1246  C       STORAGE TO THE OPERATING SYSTEM, AND THEN RECALCULATES ALL
```

```
1247  C     DYNAMIC ARRAY POINTERS.
1248  C
1249  C     ...Consolidate the fragmented memory
1250        CALL MCCOMP
1251  C     ...Return unused storage back to the operating system
1252        CALL MCGIVE
1253  C     ...Recalculate ALL pointers of the C-array
1254        DO 300 I = BCDAIDX, ECDAIDX, 1
1255        I2 = 0
1256        CALL MCFIND(CNAMES(I),I1,I2)
1257           LCPTR(I)  = I1
1258           CLENGT(I) = I2
1259   300  CONTINUE
1260  C     ...Check for dynamic memeory allocation errors
1261        IF(DEBUG)CALL MCDEBG(NOUTFL)
1262        CALL MCSTAT(MNERRS,MNUSED)
1263        IF(MNERRS.NE.0.OR.DEBUG)CALL MCLIST(NOUTFL)
1264        IF(MNERRS.NE.0.        )THEN
1265           CALL MCEROR(NOUTFL)
1266           CALL QAABORT('(MEMADJC3) - ### CHARACTER ### '//
1267       $                 'DYNAMIC MEMORY ALLOCATION ERRORS ***')
1268        ENDIF
1269        RETURN
1270  C-----------------------------------------------------------------------
1271  C**** END OF SUBROUTINE MEMADJC ****
1272  C-----------------------------------------------------------------------
1273        END
1274        SUBROUTINE PRCOUT(AAPPT,AATET,AAINFR,AAINF,AAOFF,AARUN,MASSB,
1275       +                  MAXBAL)
1276  C***********************************************************************
1277  C
1278  C***********************************************************************
1279  C
1280  C  Memo from Joe Hevesi 8 Feb 1999
1281  C
1282  C Here's the format for the main net infiltration output file:
1283  C
1284  C Output file for mapping average_annual mass balance terms (files with *.4e4
1285  C                                                            extension):
1286  C easting(i) = utm easting (m)
1287  C northing(i) = utm northing (m)
1288  C aappt(i) = average annual precip (mm/yr)
1289  C aatet(i) = average annual evapotranspiration (mm/yr)
1290  C aainfr(i) = average annual infiltration from runon (mm/yr)
1291  C aadsw(i) = average annual change in root-zone water content (mm/yr)
1292  C aainf(i) = average annual net infiltration rate (mm/yr)
1293  C aaoff(i) = average annual runoff generation (mm/yr)
1294  C aarun(i) = average annual surface water runon (mm/yr)
1295  C massb(i) = average annual water balance calculation error (mm/yr)
1296  C maxbal(i) = maximum daily water balance calculation error  (mm/day)
1297  C
1298        INCLUDE 'PARAMS.INC'
1299        INCLUDE 'POSTINFIL.INC'
1300        INCLUDE 'FILEUN.INC'
1301        INTEGER I, ISAV
1302        REAL EAST,NORTH,AAPPT(*),AATET(*),
1303       +     AAINFR(*),AADSW,AAINF(*),AAOFF(*),AARUN(*),
```

```
1304        +     MASSB(*),MAXBAL(*),PPTRAIN,PPTSNOW,SNOWCVR,SNOWMELT,SUBLIM,
1305        +     MASSBAL2,SUM_INF
1306         CHARACTER*256 HEADER
1307         LOGICAL EOF
1308   C
1309         EOF = .FALSE.
1310         I = 0
1311   C     ...read header line
1312           READ(INFLOUT1,1000) HEADER
1313    1000   FORMAT(A)
1314
1315   C*      DO WHILE (.NOT. EOF)
1316      10 CONTINUE
1317            I = I+1
1318            IF (I.GT.MAXSOLN) THEN
1319              PRINT *, 'The number of locations is greater than ',
1320        +                      'allowed (',MAXSOLN,')'
1321              CALL QAABORT('(PRCOUT) - reading INFILTRATION values ***')
1322            ENDIF
1323            READ(INFLOUT1,1001,END=99) EAST,NORTH,AAPPT(I),
1324        +         PPTRAIN,PPTSNOW,SNOWCVR,SNOWMELT,SUBLIM,
1325        +         AATET(I),AAINFR(I),AADSW,AAINF(I),AAOFF(I),AARUN(I),
1326        +         MASSB(I),MAXBAL(I),MASSBAL2
1327   1001     FORMAT(1X,2F11.1,1X,11F13.5,F13.4,4X,3E17.7)
1328   C *** Save infiltration rates for designated locations
1329            IF(EAST .LE. MAXEAST .AND. EAST .GT. MINEAST .AND.
1330        +      NORTH .LE. MAXNORTH .AND. NORTH .GT. MINNORTH)
1331        +      THEN
1332               ISAV = ISAV + 1
1333               SUM_INF = SUM_INF + AAINF(I)
1334             ENDIF
1335         GO TO 10
1336   C*      END DO
1337      99 CONTINUE
1338         IF(ISAV .GT. 0.0) THEN
1339   C *** compute mean of saved pointwise infiltration rates
1340           NUMSOLN = ISAV
1341           AVE_INF = SUM_INF/NUMSOLN_
1342           INCLUDE = .TRUE.
1343         ELSE
1344           INCLUDE = .FALSE.
1345         ENDIF
1346         WRITE (NOUTFL,*) 'This watershed has ',NUMSOLN,' points inside',
1347        +                 ' the specified boundary'
1348   C
1349         RETURN
1350         END
1351         SUBROUTINE PRCOUT2(INFLOUT2,NUMCELL,AVE_INFIL,AVE_RUNOFF)
1352   C*****************************************************************************
1353   C
1354   C
1355   C*********************************************************************
1356   C234567
1357   C
1358         IMPLICIT NONE
1359         INCLUDE 'PARAMS.INC'
1360         INCLUDE 'FFRDAT.INC'
```

```
      INTEGER IERR, INFLOUT2, IOSTAT, ISCRAT, NFIELD, NUMCELL
      INTEGER IVALUE(MFIELD), KVALUE(MFIELD)
      REAL AVE_INFIL,AVE_RUNOFF
      REAL RVALUE(MFIELD)
      CHARACTER*(NFORM) CVALUE(MFIELD)
      CHARACTER*40 LINE
      CHARACTER*80 HEADER

C<><><><><><><><><><><><>
C...BEGIN PROCEDURES...
C<><><><><><><><><><><><>

      IERR   =  0
      REWIND INFLOUT2

C     ...read header line
       READ(INFLOUT2,1000) HEADER
 1000  FORMAT(A80)

  10  CALL FFRDFLDS( INFLOUT2, ISCRAT, ' ',    MFIELD, IOSTAT,
     +               NFIELD,   KVALUE, CVALUE, IVALUE, RVALUE )
C
  20  IF(IOSTAT.LT.0)THEN
C        ...End Of File for INFIL control card input file found
         GOTO 100

      ELSEIF(IOSTAT.GT.0)THEN
C        ...Set the error flag, abort after EOF found
         IERR = IERR + 1
C        ...Read next record
         GOTO 10

      ELSEIF(IOSTAT.EQ.0)THEN

         IF((KVALUE(1).EQ.-1).OR.(CVALUE(1)(:1).EQ.'!').OR.
     +      (NFIELD.EQ.0))THEN
C           ...This is a comment line without information
C           ...Read next record
            GOTO 10

         ELSEIF( CVALUE(1)(:5).EQ.'TOTAL' .AND.
     +           CVALUE(4)(:3).EQ.'LOC' ) THEN
C****    this is the number of active cells in the watershed
            NUMCELL = IVALUE(5)
         ELSEIF(CVALUE(1)(:3).EQ.'NET')THEN
C****    this is the mean infiltration over this watershed
            AVE_INFIL = RVALUE(3)
         ELSEIF(CVALUE(1)(:3).EQ.'RUN')THEN
C****    this is the mean surface runoff over this watershed
            AVE_RUNOFF = RVALUE(3)
         ENDIF

      ENDIF
      GO TO 10

  100 CONTINUE
C
```

```
      RETURN
      END
*DECK PREPRO
      SUBROUTINE PREPRO( FIDB, FTXT, FTRN, FTRN2, FODB )
C*********************************************************************
C***                                                               ****
C***          P_R_E - P_R_O_cessing program execution module       ****
C***          - - -   - - -                                        ****
C*********************************************************************
C
C  PURPOSE:     PROGRAM input preprocessor which performs:
C               1.  QA functions
C               2.  Uses standard routines from CAMCON_LIB
C
C  AUTHOR:      Ron McCurley
C
C  UPDATED:     19-JAN-1999  --RDM: First Ed. for POSTINFIL
C               19-AUG-1999  --RDM: modified for adding INFIL summary output
C
C  CALLED BY:   A1MAIN {main program}
C
C  CALLS:       QASETUP
C               FILCMDLINN
C               QABANNER
C               FILDFNAM
C               WRFILN
C               QAABORT
C               FILRDNAMS
C               FILOPEN
C               IQAERRUNI
C               QAPAGE
C               QABANNER
C               QADOEDIS
C               FILOPEN
C               FILWRNAMS
C               FILOPEN
C
C  ARGUMENTS:
C  ENTRY/
C   --common block
C    /FILEUN/ (from FORTRAN statement: INCLUDE 'FILEUN.INC')
C     INPUTF = Device no. of input text file
C     ISCRAT = Device no. of scratch reading file
C     NOUTFL = Device no. of output diagnostics/debug file
C
C    /QA/
C     PRGNAM = Program name (CHAR*8)
C     PRGVER = Program version (CHAR*8)
C     PRGUPD = Program update (CHAR*8)
C     PRGATH = Program author (CHAR*80)
C     PRGSPN = Program sponsor (CHAR*80)
C
C  LOCAL/
C
C  EXIT/
C   --common blocks
C    /FILEUN/ (from FORTRAN statement: INCLUDE 'FILEUN.INC')
```

```
1475  C     NOUTFL = Device no. of DIAGNOSTICS/DEBUG output file
1476  C
1477  C    /TOOLS/  (from FORTRAN statement: INCLUDE 'TOOLS.INC')
1478  C
1479  C   --through subroutine call
1480  C     FIDB   = Filename of input CAMDAT
1481  C     FTXT   = Filename of POSTINFIL input text file
1482  C     FTRN   = Filename of INFIL pointwise output transfer file
1483  C     FTRN2  = Filename of INFIL summary output transfer file
1484  C     FODB   = Filename of output CAMDAT
1485  C
1486  C*********************************************************************
1487  C234567
1488        IMPLICIT NONE
1489        INCLUDE 'FILEUN.INC'
1490        INCLUDE 'QA.INC'
1491        INCLUDE 'TOOLS.INC'
1492        INTEGER  I,   IERR,   I1,  I2,   MAXFLD, NUMFLD
1493        PARAMETER(MAXFLD = 6)
1494        CHARACTER*(*)   FIDB, FODB, FTRN, FTRN2, FTXT
1495        CHARACTER*(132) FILESP(MAXFLD), FIELDS(MAXFLD)
1496  C
1497  C<><><><><><><><><><><><>
1498  C...BEGIN PROCEDURES...
1499  C<><><><><><><><><><><><>
1500  C
1501        IERR   = 0
1502        NOUTFL = 0
1503
1504  C     ...Load the /PROCQA/ common with information
1505        CALL QASETUP(PRGNAM,PRGVER,PRGUPD,PRGATH,PRGSPN)
1506
1507  C     ...Write program banner to the terminal screen
1508        CALL QABANNER(0,
1509       $'Program for postprocessing infiltration maps',
1510       $' ',' ')
1511
1512  C     ...Check command line for a BATCH run
1513        DO 10 I=1,MAXFLD
1514           FIELDS(I)(:132)=' '
1515     10 CONTINUE
1516        CALL FILCMDLIN(MAXFLD,NUMFLD,FIELDS)
1517        IF(NUMFLD.GT.MAXFLD)THEN
1518        PRINT*,'*****************************************************'
1519        PRINT*,'*** Too many input files specified on command line ***'
1520        PRINT*,'*****************************************************'
1521        ENDIF
1522
1523  C     ...Define the files for FILRDNAMS
1524        CALL FILDFNAM('Input CAMDAT','inp','req',
1525       $              'POSTLHS.CDB')
1526        CALL FILDFNAM('Input Text Data File','inp','opt',
1527       $              'POSTINFIL.INP')
1528        CALL FILDFNAM('Output INFIL Summary Transfer File','inp','opt',
1529       $              'INFILSUM.OUT')
1530        CALL FILDFNAM('Output INFIL Point Transfer File','inp','opt',
1531       $              'INFILPT.OUT')
```

```
      CALL FILDFNAM('Output CAMDAT','out','req',
     $              'POSTINFIL.CDB')
      CALL FILDFNAM('Output Diagnostics/Debug File','out','opt',
     $              'POSTINFIL.DBG')

C     ...Read in the names of the files
      CALL FILRDNAMS(FILESP,IERR)
      IF(IERR.NE.0)THEN
         CALL FILWRNAMS(0,FILESP)
         CALL QAABORT('(PREPRO) - Incorrect file assignments ***')
      ENDIF

C     ...Open the Diagnostics/debug output file
      IF(FILESP(MAXFLD).NE.' ')THEN
         NOUTFL    = 7
         OPEN(NOUTFL,FILE=FILESP(MAXFLD),STATUS='UNKNOWN',
     $        FORM='FORMATTED',RECL=256)
      ELSE
         NOUTFL    = 0
         CALL FILOPEN(NOUTFL,'OUT','FORM','SYS$OUTPUT',IERR)
      ENDIF

C     ...Set the DEVICE number of the Diagnostics/debug output file
      CALL IQAERRUNI(NOUTFL)

      IF(NOUTFL.GT.0)THEN
C        ...Write headers, banner, DOE disclaimer, and file assignments
         CALL QAPAGE(NOUTFL,' ')
         CALL QABANNER(NOUTFL,
     $'Program for postprocessing infiltration maps',
     $' ',' ')
         CALL QADOEDIS(NOUTFL,'*')
         CALL FILWRNAMS(NOUTFL,FILESP)
      ENDIF

      FIDB  = FILESP(1)(:132)
      FTXT  = FILESP(2)(:132)
      FTRN2 = FILESP(3)(:132)
      FTRN  = FILESP(4)(:132)
      FODB  = FILESP(5)(:132)

      RETURN
C-----------------------------------------------------------------------
C**** END OF SUBROUTINE PREPRO ****
C-----------------------------------------------------------------------
      END
*DECK PROCDB
      SUBROUTINE PROCDB( MAXC, MAXQ, IQ, LQ, Q, C )
C***********************************************************************
C****                                                                ****
C****          P_R_O_C_ess  input D_ata  B_ase  routine              ****
C****          - - - -            -      -                           ****
C***********************************************************************
C
C  PURPOSE:    Process dynamic memory allocation for input
C              CAMDAT file etc...
C
```

```
1589  C  AUTHOR:      Jonathan S. Rath
1590  C
1591  C  UPDATED:     13-JUN-1994  --JSR: First Ed. for WPRD.
1592  C
1593  C  CALLED BY:   WPRD {main program}
1594  C
1595  C  CALLS:       DBISIZES
1596  C               DBITITLE
1597  C               ADDMEM
1598  C               DYNMEM
1599  C               ZERODA
1600  C               DYNMEMC
1601  C               RECHEA
1602  C
1603  C  ARGUMENTS:
1604  C  ENTRY/
1605  C   --common blocks
1606  C    /FILEUN/ ($include 'FILEUN.INC')
1607  C     IDB    = Device no of Input CAMDAT File
1608  C     ODB    = "            Output CAMDAT file
1609  C
1610  C   --through subroutine call
1611  C     MAXC   = Total no. of CHAR*1 words allocated to dynamic storage
1612  C     MAXQ   = Total no. of words allocated to dynamic storage
1613  C     IQ     = INTEGER Dynamic Q-array (equivalenced to Q)
1614  C     LQ     = LOGICAL Dynamic Q-array (equivalenced to Q)
1615  C     Q      = REAL Dynamic memory base array
1616  C     C      = CHARACTER Dynamic memory base array
1617  C
1618  C  LOCAL/
1619  C     IERR   = INTEGER error code
1620  C     MORMEMC= No, additional words to be allocated to C array
1621  C     MORMEMR= No. additoinal words to be allocated to Q array
1622  C
1623  C  EXIT/
1624  C   --common blocks
1625  C    /LOCATR/ (from FORTRAN statement: INCLUDE 'LOCAT.INC')
1626  C     L[name]= Pointer to [name] Q-array integers
1627  C     LRPTR  = Pointer of Q-array
1628  C
1629  C    /LOCATC/ (from FORTRAN statement: INCLUDE 'LOCAT.INC')
1630  C     L[name]= Pointer to [name] C-array integers
1631  C     LCPTR  = Pointer of C-array
1632  C
1633  C    /GEOPAR/ (from FORTRAN statement: INCLUDE 'GEOPAR.INC')
1634  C     HEADER = Character description of problem, etc...
1635  C     LESSEL = List of Element Side sets concatenated Element List
1636  C     LESSNL = List of Element Side sets concatenated Nodes associated
1637  C              with each element side set Node List
1638  C     LNPSNL = List of Nodal Point set concatenated Nodes List
1639  C     MAXNENO= Max. no. of nodes of any element
1640  C     NDIM   = Number of physical DIMensions
1641  C     NDBCAM = CAMDAT version number
1642  C     NELBLK = Number of ELement BLocKs in mesh
1643  C     NELX   = No. of elements along X-axis of mesh
1644  C     NELY   = No. of elements along Y-axis of mesh
1645  C     NELZ   = No. of elements along Z-axis of mesh
```

```
1646  C     NQAREC = Number of QA records
1647  C     NSTEP  = No. of time steps in input CAMDAT (INTEGER)
1648  C     NUMEL  = NUmber of ELeMents in mesh
1649  C     NUMNOD = NUMber of NODes in mesh
1650  C     NUMNPS = No. of Nodal Point Sets
1651  C     NUMESS = No. of Element Side Sets
1652  C     NUQATR = No. of UniQue ATtRibute names
1653  C     NUQPR0 = No. of UniQUe PRoperty names (old)
1654  C     NUQPRP = No. of UniQUe PRoPerty names (new)
1655  C     NVAREL = No. of ELEMENT type ANALYSIS variables
1656  C     NXINFO = No. of information records
1657  C
1658  C*********************************************************************
1659  C234567
1660        IMPLICIT NONE
1661        INCLUDE 'FILEUN.INC'
1662        INCLUDE 'LOCAT.INC'
1663        INCLUDE 'GEOPAR.INC'
1664        INTEGER  IERR, MAXC, MAXQ, MORMEMC, MORMEMR
1665        INTEGER IQ(*)
1666        REAL Q(*)
1667        CHARACTER*(1) C(*)
1668        LOGICAL LQ(*)
1669  C
1670  C<><><><><><><><><><><><>
1671  C...Begin Procedures...
1672  C<><><><><><><><><><><><>
1673  C
1674        IERR = 0
1675  C     ...Read input CAMDAT file and get array sizes
1676        CALL DBISIZES( IDB,     NQAREC,  NXINFO,  NDIM,   NUMNOD,
1677       $               NELBLK,  NUMEL,   NELX,    NELY,   NELZ,
1678       $               MAXNENO, NUQATR,  NUQPRP,  NUMNPS, LNPSNL,
1679       $               NUMESS,  LESSEL,  LESSNL,  IERR )
1680        IF(IERR.NE.0)CALL QAABORT('(PROCDB) - reading input CAMDAT '//
1681       $                          'sizing parameters ***')
1682        CALL DBITITLE( IDB, HEADER, IERR )
1683        IF(IERR.NE.0)CALL QAABORT('(PROCDB) - reading input CAMDAT '//
1684       $                          'TITLE ***')
1685
1686        CALL DBINVAR( IDB, NVARHI, NVARGL, NVARND, NVAREL, IERR )
1687        IF(IERR.NE.0)CALL QAABORT('(PROCDB) - reading number of input'//
1688       $                          ' CAMDAT analysis variables ***')
1689
1690  C     ...Initialize dynamic array lengths and compute active memory
1691        CALL ADDMEM(  MORMEMR, MORMEMC )
1692        CALL DYNMEM(  MORMEMR, MAXQ, Q )
1693        CALL ZERODA( IQ, LQ )
1694        CALL DYNMEMC( MORMEMC, MAXC, C )
1695
1696  C     ...Read various CAMDAT "header" sections into arrays
1697        CALL RECHEA( Q,            C,            IQ(LIDEBLK), IQ(LNUMPRP),
1698       $             LQ(LIASPRP), Q(LXMATPR),  C(LQAINFO),  C(LNAMELB),
1699       $             C(LNMATPR) )
1700
1701        RETURN
1702  C---------------------------------------------------------------------
```

```
C**** END OF SUBROUTINE PROCDB *****
C-----------------------------------------------------------------------
      END
*DECK PROCIN
      SUBROUTINE PROCIN(NAMGLV2)
C***********************************************************************
C***                                                                 ****
C***        PROC_ess  POSTINFIL  INput control  data  file  module   ****
C***        ----                 --                                  ****
C***********************************************************************
C
C  PURPOSE:     Reads POSTINFIL input text file
C
C  AUTHOR:      Ron McCurley
C
C  UPDATED:
C
C  CALLED BY:
C
C  CALLS:       FREFLD
C               MDRSRV
C               RPSBCS
C               MDDEL
C               ABORT
C
C  ARGUMENTS:
C  ENTRY/
C   --common blocks
C    /COMMIO/ ($INCLUDE 'COMMIO.INC')
C     ISCRAT = Device no. of PREINFIL scratch file
C     NOUTFL = Device no. of diagnostics/debug output file
C     PRSUTR = "            PRE-INFIL  Control Card Data file
C
C    /FFRDAT/ ($INCLUDE 'FFRDAT.INC')
C     MFIELD = Max. no. of fields that Free-Field-Reader can process
C     NFORM  = Max. length of a CHARACTER data field
C
C    /ARRAY/ ($INCLUDE 'ARRAY.INC')
C     Q      = Dynamic memory allocation BASE array
C     MAXQ   = Length of BASE array, Q
C
C    /LOCAT/ ($INCLUDE 'LOCAT.INC')
C     L[name]= Pointer to [name] dynamic array
C
C   --through subroutine call
C     TDPROP = LOGICAL for finding TIME-DEPENDENT property data
C
C  LOCAL/
C     CVALUE = CHARACTER values of the data fields
C     IERR   = INTEGER error flag
C     IOSTAT = INTEGER value for ANSI FORTRAN I/O status
C     IVALUE = INTEGER values of the data fields
C     KVALUE = Translation states of the data fields
C     NFIELD = Number of fields
C     RVALUE = REAL values of the data fields
C
C  EXIT/
```

```
1760  C    --common block
1761  C     /POSTINFIL/ ($INCLUDE 'POSTINFIL.INC')
1762  C      TSTART = Time value when Initial Conditions are applied (REAL)
1763  C      WRITIC = LOGICAL variable for writing I.C.s to TAPE55
1764  C
1765  C    --through subroutine call
1766  C      NODEBC = INTEGER array of node points corresponding to B.C. values
1767  C
1768  C*********************************************************************
1769  C234567
1770        IMPLICIT NONE
1771        INCLUDE 'FILEUN.INC'
1772        INCLUDE 'FFRDAT.INC'
1773        INCLUDE 'POSTINFIL.INC'
1774  C*      INCLUDE 'FLAGS.INC'
1775        INTEGER IERR, IOSTAT, LTEMP, NFIELD
1776        INTEGER IVALUE(MFIELD), KVALUE(MFIELD)
1777        REAL RVALUE(MFIELD)
1778        CHARACTER*(8) NAMGLV2(*)
1779        CHARACTER*(NFORM) CVALUE(MFIELD)
1780
1781  C<><><><><><><><><><><><>
1782  C...BEGIN PROCEDURES...
1783  C<><><><><><><><><><><><>
1784
1785        IERR   =  0
1786        REWIND ICNTRLFIL
1787
1788   10   CALL FFRDFLDS( ICNTRLFIL, ISCRAT, ' ',    MFIELD, IOSTAT,
1789      +                NFIELD, KVALUE, CVALUE, IVALUE, RVALUE )
1790  C
1791   20   IF(IOSTAT.LT.0)THEN
1792  C        ...End Of File for INFIL control card input file found
1793           GOTO 100
1794
1795        ELSEIF(IOSTAT.GT.0)THEN
1796  C        ...Set the error flag, abort after EOF found
1797           IERR = IERR + 1
1798  C        ...Read next record
1799           GOTO 10
1800
1801        ELSEIF(IOSTAT.EQ.0)THEN
1802
1803           IF((KVALUE(1).EQ.-1).OR.(CVALUE(1)(:1).EQ.'!').OR.
1804      +       (NFIELD.EQ.0))THEN
1805  C           ...This is a comment line without information
1806  C           ...Read next record
1807              GOTO 10
1808
1809           ELSEIF(CVALUE(1)(:6).EQ.'*ARRAY')THEN
1810  C           ...Read INFIL ARRAY data
1811  C*            CALL RDARRD( NOUTFL, IOSTAT, NFIELD, KVALUE,
1812  C*     1                   CVALUE, IVALUE, RVALUE )
1813              GO TO 10
1814           ELSEIF(CVALUE(1)(:6).EQ.'*STORE')THEN
1815  C****    read input for analysis array where infiltration rates, etc. reside
1816              CALL RSTORE( NOUTFL, IOSTAT, NFIELD, KVALUE,
```

```
1817        1                    CVALUE, IVALUE, RVALUE )
1818              GO TO 10
1819
1820           ELSEIF(CVALUE(1)(:4).EQ.'*END')THEN
1821  C           ...Found END of input control data
1822              GOTO 100
1823
1824           ELSE
1825  C           ...Found meaningless data
1826  C           ...Read next record
1827              GOTO 10
1828           ENDIF
1829        ENDIF
1830
1831   100  IF(IERR.GT.0)CALL QAABORT('(PROCIN) - Read ERRORS encountered '//
1832       +                        'reading PREINFIL input text file ***')
1833        NAMGLV2(1) = NAMNUM
1834        NAMGLV2(2) = NAMINF
1835        NAMGLV2(3) = NAMRUN
1836  C
1837        RETURN
1838  C-----------------------------------------------------------------------
1839  C**** END OF SUBROUTINE PROCIN ****
1840  C-----------------------------------------------------------------------
1841        END
1842  *DECK RECHEA
1843        SUBROUTINE RECHEA( Q,       C,       IDEBLK, NUMPRP, IASPRP,
1844       $                   XMATPR, QAINFO, NAMELB, NMATPR )
1845  C***********************************************************************
1846  C****                                                                ****
1847  C****          R_E_ad  C_amdat  H_E_A_der  section  routine          ****
1848  C****          - -     -        - - -                                ****
1849  C***********************************************************************
1850  C
1851  C  PURPOSE:    Reads pertinent "HEADER" sections of the input
1852  C              CAMDAT file into common blocks and arrays
1853  C
1854  C  AUTHOR:     Jonathan S. Rath
1855  C
1856  C  UPDATED:    13-JUN-1994  --JSR: First Edition
1857  C
1858  C  CALLED BY:  PROCDB
1859  C
1860  C  CALLS:      DBIELBLK
1861  C              DBIPROP
1862  C              DBIPRNM
1863  C              DBIQAREC
1864  C              DBINSTEP
1865  C              QAABORT
1866  C
1867  C  ARGUMENTS:
1868  C  ENTRY/
1869  C   --common blocks
1870  C    /FILEUN/ (from FORTRAN statement: INCLUDE 'FILEUN.INC')
1871  C     IDB    = Device no of Input CAMDAT File
1872  C     ODB    = "            Output CAMDAT file
1873  C
```

```
1874  C     /GEOPAR/ (from FORTRAN statement: INCLUDE 'GEOPAR.INC')
1875  C      NELBLK = Number of ELement BLocKs in mesh (i.e. # of materials)
1876  C      NQAREC = Number of QA records
1877  C      NSTEP  = No. of time steps in input CAMDAT (INTEGER)
1878  C      NVAREL = No. of element variables (INTEGER)
1879  C      NUQPRP = No. of UniQUe PRoperty names (INTEGER)
1880  C
1881  C    --through subroutine call
1882  C      Q      = REAL type base dynamic array
1883  C      C      = CHARACTER*1 type base dynamic array
1884  C
1885  C   LOCAL/
1886  C      IERR   = INTEGER error code
1887  C
1888  C   EXIT/
1889  C    --through subroutine call
1890  C      IDEBLK = INTEGER ID array of matrial regions (INTEGER)
1891  C      NUMPRP = No. of properties per material region (INTEGER)
1892  C      IASPRP = Logical array for properties in materials
1893  C      XMATPR = Properties array (REAL)
1894  C      QAINFO = Quality Assurance array (CHAR*8)
1895  C      NAMELB = Names of the material regions (CHAR*8)
1896  C      NMATPR = Names of the properties (CHAR*8)
1897  C
1898  C*********************************************************************
1899  C234567
1900        IMPLICIT NONE
1901        INCLUDE 'FILEUN.INC'
1902        INCLUDE 'GEOPAR.INC'
1903        INTEGER  I, IERR, J, K, NHISTP
1904        INTEGER IDEBLK(*), NUMPRP(*)
1905        REAL Q(*), XMATPR(*)
1906        CHARACTER*(1) C(*)
1907        CHARACTER*(8) NAMELB(*), NMATPR(*), QAINFO(*)
1908        LOGICAL IASPRP(*)
1909  C
1910  C<><><><><><><><><><><><><>
1911  C...Begin Procedures...
1912  C<><><><><><><><><><><><><>
1913  C
1914        IERR = 0
1915
1916  C     ...Read material ID and names information
1917        CALL DBIELBLK( IDB, NELBLK, NAMELB, IDEBLK, IERR )
1918        IF(IERR.NE.0)
1919       $CALL QAABORT('(RECHEA) - reading material ID and names array ***')
1920
1921  C     ...Read property names
1922        CALL DBIPRNAM( IDB, NUQPRP, NMATPR, Q, C, IERR )
1923        IF(IERR.NE.0)
1924       $CALL QAABORT('(RECHEA) - reading PROPERTY NAMES array ***')
1925
1926  C     ...Read property data
1927        DO 10 I=1,NUQPRP
1928           J = (I-1)*NELBLK+1
1929           CALL DBIPROP( IDB, '*', I, IASPRP(J), XMATPR(J),    $
1930  Q,        C, IERR )
```

```
1931            IF(IERR.NE.0)
1932         $  CALL QAABORT('(RECHEA) - reading PROPERTY values ***')
1933      10 CONTINUE
1934
1935   C     ...Construct the number of properties per material array
1936         DO 30 J=1,NELBLK
1937            NUMPRP(J) = 0
1938            DO 20 I=1,NUQPRP
1939               K = (I-1)*J+1
1940               IF(IASPRP(K))NUMPRP(J) = NUMPRP(J)+1
1941      20    CONTINUE
1942      30 CONTINUE
1943
1944   C     ...Read QA data
1945         CALL DBIQAREC( IDB, NQAREC, QAINFO, IERR )
1946         IF(IERR.NE.0)CALL QAABORT('(RECHEA) - reading QA array ***')
1947
1948   C     ...Get the number of time steps (NSTEP)
1949         CALL DBINSTEP( IDB, NSTEP, NHISTP, IERR )
1950         IF(IERR.NE.0)CALL QAABORT('(RECHEA) - obtaining input CAMDAT '//
1951        $                          'no. of time steps ***')
1952
1953         RETURN
1954   C-----------------------------------------------------------------------
1955   C**** END OF SUBROUTINE RECHEA *****
1956   C-----------------------------------------------------------------------
1957         END
1958   *DECK RSTORE
1959         SUBROUTINE RSTORE( DBGFIL, IOSTAT, NFIELD, KVALUE,
1960        1                   CVALUE, IVALUE, RVALUE )
1961   C***********************************************************************
1962   C****                                                                ****
1963   C****         R_ead  S_T_O_R_E  parameters  module                 ****
1964   C****         -    -  - - - -                                      ****
1965   C***********************************************************************
1966   C
1967   C   PURPOSE:     Reads STORE parameters for INFIL from the
1968   C                POSTINFIL batch input file
1969   C
1970   C   AUTHOR:      Ron McCurley
1971   C
1972   C   UPDATED:
1973   C
1974   C  CALLED BY:    PROCIN
1975   C
1976   C  CALLS:        FREFLD
1977   C                ABORT
1978   C
1979   C  ARGUMENTS:
1980   C  ENTRY/
1981   C    --common blocks
1982   C     /COMMIO/ ($INCLUDE 'COMMIO.INC')
1983   C      ISCRAT = Device no. of PREINFIL scratch file
1984   C      FILEIN = Device no. of PREINFIL input text file
1985   C      DBGFIL = Device no. of diagnostics/debut output file
1986   C
1987   C     /POSTINFIL/ ($INCLUDE 'POSTINFIL.INC')
```

```
1988  C       NFILES - number of input CDBs
1989  C       TIMINT - time of intrusion into repository(1000,3000,5000,7000,
1990  C                or 9000 years)
1991  C       ROOT   - root of variable, 3 characters
1992  C                (ie AC5 if AC5u234,ACPU240,... are wanted)
1993  C       PATHWAY- Pathway to MAN ; may be 'STOCK_POnd'(default),
1994  C                'FARM_FAMily',or 'DRILler'
1995  C
1996  C    --through subroutine call
1997  C      MFIELD = Max. no. of data fields FFR can process
1998  C      IOSTAT = ANSI FORTRAN I/O error flag
1999  C      NFIELD = No. of data fields read by FFR
2000  C      KVALUE = INTEGER array of types of data fields read by FFR.
2001  C      CVALUE = CHARACTER  array of data fields read by FFR.
2002  C      IVALUE = INTEGER    "
2003  C      RVALUE = REAL       "
2004  C
2005  C  LOCAL/
2006  C      none
2007  C
2008  C  EXIT/
2009  C    --common blocks
2010  C     /INFIL/ ($INCLUDE 'INFIL.INC')
2011  C      ...REAL variables
2012  C
2013  C    --through subroutine call
2014  C      MFIELD = Max. no. of data fields FFR can process
2015  C      IOSTAT = ANSI FORTRAN I/O error flag
2016  C      NFIELD = No. of data fields read by FFR
2017  C      KVALUE = INTEGER array of types of data fields read by FFR.
2018  C      CVALUE = CHARACTER  array of data fields read by FFR.
2019  C      IVALUE = INTEGER    "
2020  C      RVALUE = REAL       "
2021  C
2022  C*********************************************************************
2023  C234567
2024        IMPLICIT NONE
2025        INCLUDE 'FFRDAT.INC'
2026        INCLUDE 'GEOPAR.INC'
2027        INCLUDE 'POSTINFIL.INC'
2028  C*      INCLUDE 'FLAGS.INC'
2029
2030        INTEGER I, IERR, IOSTAT, ISAV, J, N, NFIELD, DBGFIL
2031        INTEGER IVALUE(*), KVALUE(*)
2032        REAL RVALUE(*)
2033        CHARACTER*(NFORM) CVALUE(*)
2034
2035  C<><><><><><><><><><><><><>
2036  C...BEGIN PROCEDURES...
2037  C<><><><><><><><><><><><><>
2038  C
2039        IERR=0
2040        DO 40 I=2,NFIELD
2041           IF(CVALUE(I)(:5).EQ.'ARRAY')THEN
2042              ARRAY_STORE= CVALUE(I+1)
2043  C             STORE = .TRUE.
2044           ELSEIF(CVALUE(I)(:4).EQ.'TIME')THEN
```

```
2045              TIME_STORE = RVALUE(I+1)
2046           ELSEIF(CVALUE(I)(:4).EQ.'STEP')THEN
2047              STEP_STORE = RVALUE(I+1)
2048           ELSEIF(CVALUE(I)(:4).EQ.'COPY')THEN
2049              NVARGL = IVALUE(I+1)
2050           ELSEIF(CVALUE(I)(:4).EQ.'NAME') THEN
2051              ISAV = I
2052              DO 20 J=ISAV+1,NFIELD
2053                IF(CVALUE(J)(:6).EQ.'MEAN_I')THEN
2054                  NAMINF = CVALUE(J+1)
2055                  NVARGL2 = NVARGL2 + 1
2056                ELSEIF(CVALUE(J)(:3).EQ.'NUM')THEN
2057                  NAMNUM=CVALUE(J+1)
2058                  NVARGL2 = NVARGL2 + 1
2059                ELSEIF(CVALUE(J)(:6).EQ.'MEAN_R')THEN
2060                  NAMRUN=CVALUE(J+1)
2061                  NVARGL2 = NVARGL2 + 1
2062                ENDIF
2063   20         CONTINUE
2064  C
2065  C*          ELSE
2066  C*            IF(CVALUE(I).NE.BLANK) THEN
2067  C             ...Found meaningless data
2068  C*              WRITE(DBGFIL,*)' *** BAD PATHWAY KEYWORD = ',
2069  C*     +                  CVALUE(1)(:INDEX(CVALUE(1),' ')),' ***'
2070  C*              IERR = IERR + 1
2071  C*            ENDIF
2072           ENDIF
2073   40  CONTINUE
2074  100  IF(IERR.GT.0)CALL QAABORT('(RSTORE) - Errors reading pathway'//
2075      +                          ' parameters ***')
2076       RETURN
2077  C-----------------------------------------------------------------------
2078  C**** END OF SUBROUTINE RSTORE  ****
2079  C-----------------------------------------------------------------------
2080       END
2081  *DECK SUCONS
2082       SUBROUTINE SUCONS
2083  C***********************************************************************
2084  C****                                                                ****
2085  C****            S_et  U_p   C_O_N_S_tants  module                   ****
2086  C****            -     -     - - - -                                ****
2087  C***********************************************************************
2088  C
2089  C     PURPOSE:    Computes mathematical and other constants that
2090  C                 cannot be iInitialized using a block data statement.
2091  C
2092  C     AUTHOR:     Jonathan Scott Rath, Esq. [NMERI]
2093  C
2094  C     UPDATED:    18-AUG-1995  --JSR: Adapted for DOUBLE PRECSION
2095  C                 31-AUG-1993  --JSR: First Edition
2096  C
2097  C     ARGUMENTS:
2098  C     ENTRY/
2099  C       none
2100  C
2101  C     LOCAL/
```

```
C       none
C
C     EXIT/
C      --through common blocks:
C       /CONSTA/ (from FORTRAN statement: INCLUDE 'CONSTA.INC')
C        SECPYR = No. of seconds per year [DOUBLE PRECISION]
C        YRPSEC = No. of years per second [DOUBLE PRECISION]
C
C*********************************************************************
C234567
      IMPLICIT NONE
      INCLUDE 'CONSTA.INC'
C
C<><><><><><><><><><><><>
C...Begin Procedures...
C<><><><><><><><><><><><>
C
C     ...Compute pi
      PI = 4.0*ATAN(1.0)
C     ...Compute number of seconds in one year
      SECPYR = 365.24D0*24.0D0*3600.0D0
C     ...Compute years per second
      YRPSEC = 1.0D0/SECPYR
C
      RETURN
C---------------------------------------------------------------------
C**** END OF SUBROUTINE SUCONS ****
C---------------------------------------------------------------------
      END
*DECK WRCHEA
      SUBROUTINE WRCHEA( Q, CH, QAINFO, BLKNAM, SAVNAM )
C*********************************************************************
C***                                                               ****
C***        W_R_ite   C_amdat  H_E_A_der  section  module          ****
C***        - -       -        - - -                               ****
C*********************************************************************
C
C  PURPOSE:       Writes the "header" section of CAMDAT to the scratch
C                 CAMDAT file using CAMDAT_LIB routines.
C
C  AUTHOR:        Jonathan S. Rath
C
C  UPDATED:       20-SEP-1994  --JSR: Added history variable "TOTCURIE"
C                 13-JUN-1994  --JSR: First Ed. for WPRD.
C
C  CALLED BY:     WPRD {main program}
C
C  CALLS:         DBOHEA
C                 DBOQAREC
C                 DBOVRNAM
C                 QAABORT
C
C  ARGUMENTS:
C  ENTRY/
C    --common blocks
C     /FILEUN/ ($include 'FILEUN.INC')
C      IDB    = Device number of Input CAMDAT file
```

```
2159  C       ODB    = Device number of Output CAMDAT file
2160  C
2161  C      /GEOPAR/ ($include 'GEOPAR.INC')
2162  C       NQAREC = no. of QA records
2163  C
2164  C      /MISCDAT/ (from FORTRAN statement: INCLUDE 'MISCDAT.INC')
2165  C       NNUC   = No. of radionuclides used by WPRD (INTEGER)
2166  C       NRADPR = No. of RADdionuclides to be PRinted (saved and
2167  C                stored) to the output CAMDAT and debug/diagnostics
2168  C                file (INTEGER)
2169  C
2170  C     --through subroutine call
2171  C       Q      = REAL Dynamic memory base array
2172  C       CH     = CHARACTER*1 Dynamic memory base array
2173  C       QAINFO = QA information array [CHARACTER*8]
2174  C       SAVNAM = Names of the regions used in POSTINFIL,
2175  C                and to be written to output CAMDAT (CHARACTER*8)
2176  C
2177  C  EXIT/
2178  C       none
2179  C
2180  C*********************************************************************
2181  C234567
2182        IMPLICIT NONE
2183        INCLUDE 'FILEUN.INC'
2184        INCLUDE 'GEOPAR.INC'
2185        INCLUDE 'MISCDAT.INC'
2186        INCLUDE 'PARAMS.INC'
2187  C*      INCLUDE 'POSTINFIL.INC'
2188        INTEGER I, IDUM1, IERR
2189        REAL Q(*)
2190        CHARACTER*(1) CH(*)
2191        CHARACTER*(8) NAMTMP
2192        CHARACTER*(8) BLKNAM(*), QAINFO(*), SAVNAM(*)
2193  C
2194  C<><><><><><><><><><><><>
2195  C...BEGIN PROCEDURES...
2196  C<><><><><><><><><><><><>
2197  C
2198        IDUM1  = 0
2199        IERR   = 0
2200  C
2201  C**** Write QA records to output CAMDAT scratch file ****
2202  C
2203        CALL DBOQAREC(ODB,'ADD',QAINFO((NQAREC-2)*4+1),IERR)
2204        IF(IERR.NE.0)CALL QAABORT('(WRCHEA) - Trouble writing '//
2205       $'WPRD SDB QA record to CDB-scratch file %%%')
2206
2207        CALL DBOQAREC(ODB,'ADD',QAINFO((NQAREC-1)*4+1),IERR)
2208        IF(IERR.NE.0)CALL QAABORT('(WRCHEA) - Trouble writing '//
2209       $'WPRD QA record to CDB-scratch file %%%')
2210  C
2211  C**** Write all "HEADER" information to the output CAMDAT ****
2212  C
2213        CALL DBOHEAD( ODB, Q, CH, IERR )
2214        IF(IERR.NE.0)CALL QAABORT('(WRCHEA) - Writing "HEADER" '//
2215       $'section to the final output CAMDAT file %%%')
```

```
2216  C
2217  C
2218  C      ...cell infiltrations [mm/yr]
2219  C*       IF(NUMCELL.LT.MAXSOLN)THEN
2220           IF(GLOBAL) THEN
2221  C       ...Net Infiltration variable names
2222  C*           DO 30 I=1,NUMCELL
2223  C*             SAVNAM(I)(:8) = 'I_'//BLKNAM(I)(1:3)//BLKNAM(I)(7:7)
2224  C*  30       CONTINUE
2225  C         ...Infiltration in mm/yr
2226  C*           CALL DBOVRNAM( ODB,   'GLOBAL', 'ADD',  NUMCELL, SAVNAM,
2227  C*      $                   IDUM1,  Q,        CH,     IERR )
2228             IF(IERR.NE.0)CALL QAABORT('(WRCHEA) - Error writing'//
2229       $     ' NEW GLOBAL variable names (SAVNAM) to CDB %%%')
2230  C          ...Rebuild the INFILTRATION CELL names array??
2231  C*         DO 40 I=1,NUMCELL
2232  C*             SAVNAM(I)(:8) = SAVNAM(I)(3:8)//'  '
2233  C*  40      CONTINUE
2234           ELSE
2235  C
2236  C**** Write POSTINFIL CAMDAT history variable names ****
2237  C
2238  C*         DO 50 I=1,NUMCELL
2239  C*             SAVNAM(I)(:8) = 'I_'//BLKNAM(I)(:6)
2240    50      CONTINUE
2241         CALL DBOVRNAM( ODB,    'HISTORY', 'ADD',    1,   SAVNAM,
2242       $                IDUM1,   Q,         CH,      IERR )
2243         IF(IERR.NE.0)CALL QAABORT('(WRTHEA) - Error writing'//
2244       $' NEW HISTORY variable names to CDB %%%')
2245
2246  C         ...????
2247  C*         CALL DBOVRNAM( ODB,    'GLOBAL', 'ADD',   NNUC,
2248  C*      $                  BLKNAM, IDUM1,   Q,        CH,
2249  C*      $                  IERR )
2250
2251           ENDIF
2252
2253         RETURN
2254  C-----------------------------------------------------------------------
2255  C**** END OF SUBROUTINE WRCHEA *****
2256  C-----------------------------------------------------------------------
2257         END
2258  *DECK WRTANA
2259         SUBROUTINE WRTANA( Q,       C,       TIME,    BLKNAM,  NAMGLV2,
2260       $                   AVE_INFIL, INFILTR, AVE_RUNOFF )
2261  C***********************************************************************
2262  C****                                                               ****
2263  C****        W_R_i_T_e  camdat  A_N_A_lsis  section  routine        ****
2264  C****        - -  -             - - -                               ****
2265  C***********************************************************************
2266  C
2267  C  PURPOSE:     Writes analysis section of CAMDAT for a single timestep.
2268  C
2269  C  AUTHOR:      Ron McCurley
2270  C
2271  C  UPDATED:     20-JAN-1999  --RDM: First Edition for POSTINFIL
2272  C
```

```
2273  C  CALLED BY:
2274  C
2275  C  CALLS:        DBOTIME
2276  C                DBOVAR
2277  C                DBOSTEP
2278  C                QAABORT
2279  C
2280  C  ARGUMENTS
2281  C  ENTRY/
2282  C    --common blocks
2283  C     /FILEUN/ (from FORTRAN statement: 'FILEUN.INC')
2284  C      NOUTFL = Device no. of diagnostics/debut file
2285  C      ODB    = Device no. of output CAMDAT file
2286  C
2287  C     /POSTINFIL/ (from FORTRAN statement: INCLUDE 'MISCDAT.INC')
2288  C      NUMCELL = No. of INFILTRATION cells used in POSTINFIL (INTEGER)
2289  C
2290  C    --through subroutine call
2291  C      Q      = Dynamic memory REAL base array
2292  C      C      = "              CHARACTER*1 base array
2293  C      HIFLAG = History flag (REAL)
2294  C      TIME   = Time (REAL)
2295  C      BLKNAM = Names of POSTINFIL GRID blocks, array (CHARACTER*8)
2296  C      AVE_INFIL = Average infiltration rate over spatial domain of watershed
2297  C      AVE_RUNOFF= Average surface runoff rate over spatial domain of
2298  watershed
2299  C      INFILTR  = Infiltration rate in specific location, array (REAL)
2300  C
2301  C  LOCAL/
2302  C      NAMTMP = Temporary character variable (CHARACTER*8)
2303  C
2304  C  EXIT/
2305  C      none
2306  C
2307  C*********************************************************************
2308  C234567
2309        IMPLICIT NONE
2310        INCLUDE 'FILEUN.INC'
2311        INCLUDE 'GEOPAR.INC'
2312        INCLUDE 'MISCDAT.INC'
2313        INCLUDE 'POSTINFIL.INC'
2314        INTEGER I, IERR, INDX
2315        REAL AVE_INFIL, AVE_RUNOFF, DUM0, TIME, TOTINFIL, XNUMCELL
2316        REAL Q(*), INFILTR(*)
2317        CHARACTER*(1) C(*)
2318        CHARACTER*(8) NAMTMP
2319        CHARACTER*(8) BLKNAM(*), NAMGLV2(*)
2320        LOGICAL ALLVAR, WHOLE
2321        DATA GLOBAL/.TRUE./
2322  C
2323  C<><><><><><><><><><><><>
2324  C...BEGIN PROCEDURES...
2325  C<><><><><><><><><><><><>
2326        XNUMCELL = FLOAT(NUMCELL)
2327        IF(SUBSHED) THEN
2328          XNUMCELL = FLOAT(NUMSOLN)
2329          AVE_INFIL = AVE_INF
```

```
2330          ENDIF
2331    C
2332    C**** Write a time step to the CAMDAT ****
2333    C
2334          IF(HIFLAG.EQ.1)THEN
2335             WHOLE = .FALSE.
2336          ELSEIF(HIFLAG.EQ.0)THEN
2337             WHOLE = .TRUE.
2338          ENDIF
2339    C
2340    C
2341    C**** Write POSTINFIL CAMDAT history or global variables ****
2342    C
2343    C     ...Infiltration in mm/yr
2344          TOTINFIL = 0.0
2345          IF(NSTEP.GE.1) THEN
2346    C       ...Advance time step of input CDB file
2347            CALL DBISTEP( IDB, 1, TIME, ALLVAR, IERR )
2348            IF(IERR.NE.0)CALL QAABORT('(RECHEA) - obtaining input CAMDAT '//
2349         $                            'no. of time steps ***')
2350
2351            IF (GLOBAL) THEN
2352    C       ...copy global analysis variables from input CDB to output CDB
2353              CALL DBOVRNAM( ODB, 'GLOBAL', 'COPY', NVARGL, '*',
2354         +                   DUM0, Q,        C,    IERR )
2355            ENDIF
2356          ENDIF
2357
2358    C
2359    C**** Write POSTINFIL CAMDAT global variable names ****
2360    C
2361          IF (GLOBAL) THEN
2362
2363            IF(NVARGL2.GT.0)THEN
2364              CALL DBOVRNAM( ODB,    'GLOBAL', 'ADD', NVARGL2, NAMGLV2,
2365         +                   DUM0, Q,        C,    IERR )
2366              IF(IERR.GT.0)CALL QAABORT('(PRCTIM) - Error writing '//
2367         +                 'new GLOBAL variable names to output CDB ***')
2368    C     **** Write the completed time-step and advance to the next step ****
2369    C
2370              CALL DBOSTEP( ODB, Q, C, IERR )
2371              IF(IERR.NE.0)
2372         $       CALL QAABORT('(WRTANA) - Error writing a completed'//
2373         $            ' time-step and advancing to next step /DBOSTEP ***')
2374              CALL DBOTIME( ODB, TIME, WHOLE, IERR )
2375              CALL DBOVAR( ODB, 'GLOBAL', NAMGLV2(1), INDX, XNUMCELL, Q, C,
2376         +                  IERR)
2377    C         ...Mean Infiltration in mm/yr
2378              CALL DBOVAR( ODB, 'GLOBAL', NAMGLV2(2), INDX, AVE_INFIL, Q, C,
2379         +                IERR)
2380              IF(IERR.NE.0)CALL QAABORT('(WRTANA) - Error writing'//
2381         $          ' NEW GLOBAL variable values (AVE_INFIL) to CDB ***')
2382    C*          CALL DBOVAR( ODB, 'GLOBAL', NAMGLV2(3), INDX, AVE_RUNOFF, Q,
2383    C*     +                 C, IERR)
2384              IF(IERR.NE.0)CALL QAABORT('(WRTANA) - Error writing'//
2385         $           ' NEW GLOBAL variable values (AVE_RUNOFF) to CDB ***')
2386            ENDIF
```

```
2387
2388  C*      ELSE
2389  C*        IF(NVARHI2.GT.0)THEN
2390  C*           CALL DBOVRNAM( ODB,    'HISTORY', 'ADD', NVARHI2, NAMHIV2,
2391  C*     +                  DUMO, Q,          C,     IERR )
2392  C*           IF(IERR.GT.0)CALL QAABORT('(PRCTIM) - Error writing '//
2393  C*     +                  'new HISTORY variable names to output CDB ***')
2394  C*        ENDIF
2395
2396        ENDIF
2397  C
2398  C**** Write the completed time-step and advance to the next step ****
2399  C
2400        CALL DBOSTEP( ODB, Q, C, IERR )
2401        IF(IERR.NE.0)CALL QAABORT('(WRTANA) - Error writing a completed'//
2402       $' time-step and advancing to next step /DBOSTEP ***')
2403
2404        RETURN
2405  C
2406  C-----------------------------------------------------------------------
2407  C**** END OF SUBROUTINE WRTANA *****
2408  C-----------------------------------------------------------------------
2409  C
2410        END
2411  *DECK ZERODA
2412        SUBROUTINE ZERODA( IQ, LQ )
2413  C***********************************************************************
2414  C****                                                                ****
2415  C****              Z_E_R_O  D_ynamic  A_rrays  module                ****
2416  C****              - - - -  -         -                              ****
2417  C***********************************************************************
2418  C
2419  C  PURPOSE:       Sets all values of the current IQ(*) array to zero
2420  C                 and sets all values of the current LQ(*) array
2421  C                 to false.
2422  C
2423  C  AUTHOR:        Jonathan S. Rath
2424  C
2425  C  UPDATED:       13-JUN-1994  --JSR: First Ed.
2426  C
2427  C  CALLED BY:     PROCDB
2428  C                 PROCIN
2429  C
2430  C  ARGUMENTS:
2431  C  ENTRY/
2432  C    --through common blocks:
2433  C     /DYNAME/ ($include 'DYNAME.INC')
2434  C      DTYPES  = TYPES of dynamic arrays (CHARACTER*8)
2435  C
2436  C     /LENGTH/ ($include 'LENGTH.INC')
2437  C      RLENGT  = Length of dynamic arrays (INTEGER)
2438  C
2439  C     /INDEX/  ($include 'INDEX.INC')
2440  C      BRDAIDX = Beginning current dynamic array index (INTEGER)
2441  C      ERDAIDX = Ending current dynamic aray index (INTEGER)
2442  C
2443  C     /LOCAT/ ($include 'LOCAT.INC')
```

```
2444  C       LRPTR   = Dynamic array pointers array (INTEGER)
2445  C
2446  C     --through subroutine call
2447  C       IQ      = INTEGER portion of the real dynamic base array
2448  C       LQ      = LOGICAL portion of the real dynamic base array
2449  C
2450  C  LOCAL/
2451  C    none
2452  C
2453  C  EXIT/
2454  C     --through subroutine call
2455  C       IQ      = INTEGER portion of the real dynamic base array
2456  C       LQ      = LOGICAL portion of the real dynamic base array
2457  C
2458  C*********************************************************************
2459  C234567
2460        IMPLICIT NONE
2461        INCLUDE 'DYNAME.INC'
2462        INCLUDE 'LENGTH.INC'
2463        INCLUDE 'LOCAT.INC'
2464        INCLUDE 'INDEX.INC'
2465        INTEGER I, IQ(*), J
2466        LOGICAL LQ(*)
2467  C
2468  C<><><><><><><><><><><><>
2469  C...Begin Procedures...
2470  C<><><><><><><><><><><><>
2471  C
2472        DO 30 I=BRDAIDX,ERDAIDX,1
2473           IF(DTYPES(I)(:7).EQ.'INTEGER')THEN
2474  C           ...Dynamic array index is an INTEGER type
2475              DO 10 J=1,RLENGT(I)
2476                 IQ(LRPTR(I)+J-1) = 0
2477    10        CONTINUE
2478           ELSEIF(DTYPES(I)(:7).EQ.'LOGICAL')THEN
2479  C           ...Dynamic array index is a LOGICAL type
2480              DO 20 J=1,RLENGT(I)
2481                 LQ(LRPTR(I)+J-1) = .FALSE.
2482    20        CONTINUE
2483           ENDIF
2484    30  CONTINUE
2485        RETURN
2486  C
2487  C---------------------------------------------------------------------
2488  C**** END OF SUBROUTINE ZERODA ****
2489  C---------------------------------------------------------------------
2490  C
2491        END
2492  *COMDECK CONSTA
2493  C*********************************************************************
2494  C     /CONSTA/ (from FORTRAN statement: INCLUDE 'CONSTA.INC')
2495  C      AVOGAD = Avogadro's number [DOUBLE PRECISION]
2496  C      ACCNST = Activity Conversion CoNStanT [DOUBLE PRECISION]
2497  C               A radioisotope's activity conversion is equal
2498  C               to ACCNST/(half_life*Atomic_weight)
2499  C      PI     = Common mathematical constant "PI" [DOUBLE PRECISION]
2500  C      SECPYR = No. of seconds per year [DOUBLE PRECISION]
```

```
2501  C       YRPSEC = No. of years per second [DOUBLE PRECISION]
2502  C*********************************************************************
2503  C234567
2504        DOUBLE PRECISION ACCNST, AVOGAD, PI, SECPYR, YRPSEC
2505        COMMON /CONSTA/ ACCNST, AVOGAD, PI, SECPYR, YRPSEC
2506  C*********************************************************************
2507
2508  *COMDECK DYNAME
2509  C*********************************************************************
2510  C       /DYNAME/ ($include 'DYNAME.INC')
2511  C        CNAMES = Names of CHARACTER dynamic arrays (CHARACTER*8)
2512  C        DTYPES = Types of REAL dynamic arrays (CHARACTER*8)
2513  C        RNAMES = Names of REAL data dynamic arrays (CHARACTER*8)
2514  C*********************************************************************
2515  C234567
2516        CHARACTER*(8) CNAMES(6), DTYPES(6), RNAMES(6)
2517        COMMON /DYNAME/ CNAMES, DTYPES, RNAMES
2518  C*********************************************************************
2519
2520  *COMDECK FFRDAT
2521  C*********************************************************************
2522  C    /FFRDAT/ (from FORTRAN statement: INCLUDE 'FFRDAT.INC')
2523  C     NFORM   = Maximum CHARACTER string length of a data field
2524  C     MFIELD  = Maximum number of data of fields processed
2525  C*********************************************************************
2526  C234567
2527        INTEGER MFIELD, NFORM
2528        PARAMETER( NFORM = 80, MFIELD=500 )
2529  C*********************************************************************
2530
2531  *COMDECK FILEUN
2532  C*********************************************************************
2533  C     /FILEUN/ (from FORTRAN statemetn: INCLUDE 'FILEUN.INC')
2534  C      IDB       = Device no. of input CAMDAT file
2535  C      IDBSCR    = Device no. of CAMDAT scratch file
2536  C      ICNTRLFIL = Device no. of input text file
2537  C      ISCRAT    = Device no. of scratch reading file
2538  C      INFLOUT1  = Device no. of INFIL output file with pointwise
2539  C                  infiltration maps
2540  C      INFLOUT2  = Device no. of INFIL output file with summary
2541  C                  infiltration maps
2542  C      NOUTFL = Device no. of output diagnostics/debug file
2543  C      ODB    = Device no. of output CAMDAT file
2544  C*********************************************************************
2545  C234567
2546        INTEGER IDB,    IDBSCR, ICNTRLFIL, ISCRAT, INFLOUT1, INFLOUT2,
2547       $        NOUTFL, ODB
2548        COMMON /FILEUN/ IDB,      IDBSCR, ICNTRLFIL, ISCRAT, INFLOUT1,
2549       $                INFLOUT2, NOUTFL, ODB
2550  C*********************************************************************
2551
2552  *COMDECK FLWCHR
2553  C
2554  C WPRD-+-SUCONS
2555  C      |
2556  C      +-PREPRO-+-(QASETUP)
2557  C      |        |
```

```
2558  C        |          +-(QABANNER)
2559  C        |          |
2560  C        |          +-(FILCMDLIN)
2561  C        |          |
2562  C        |          +-(FILDFNAM)
2563  C        |          |
2564  C        |          +-(FILDFNAM)
2565  C        |          |
2566  C        |          +-(FILDFNAM)
2567  C        |          |
2568  C        |          +-(FILDFNAM)
2569  C        |          |
2570  C        |          +-(FILDFNAM)
2571  C        |          |
2572  C        |          +-(FILDFNAM)
2573  C        |          |
2574  C        |          +-(FILDFNAM)
2575  C        |          |
2576  C        |          +-(FILDFNAM)
2577  C        |          |
2578  C        |          +-(FILDFNAM)
2579  C        |          |
2580  C        |          +-(FILDFNAM)
2581  C        |          |
2582  C        |          +-(FILDFNAM)
2583  C        |          |
2584  C        |          +-(FILRDNAMS)
2585  C        |          |
2586  C        |          +-(FILWRNAMS)
2587  C        |          |
2588  C        |          +-(QAABORT)
2589  C        |          |
2590  C        |          +-(FILOPEN)
2591  C        |          |
2592  C        |          +-(FILOPEN)
2593  C        |          |
2594  C        |          +-(FILOPEN)
2595  C        |          |
2596  C        |          +-(FILOPEN)
2597  C        |          |
2598  C        |          +-(FILOPEN)
2599  C        |          |
2600  C        |          +-(FILOPEN)
2601  C        |          |
2602  C        |          +-(IQAERRUNI)
2603  C        |          |
2604  C        |          +-(QAPAGE)
2605  C        |          |
2606  C        |          +-(QABANNER)
2607  C        |          |
2608  C        |          +-(QADOEDIS)
2609  C        |          |
2610  C        |          +-(FILWRNAMS)
2611  C        |
2612  C        +-(MDINIT)
2613  C        |
2614  C        +-(MCINIT)
```

```
2615  C      |
2616  C      +-(DBSETUP)
2617  C      |
2618  C      +-(QAABORT)
2619  C      |
2620  C      +-(DBERRUNI)
2621  C      |
2622  C      +-(DBIOPEN)
2623  C      |
2624  C      +-(QAABORT)
2625  C      |
2626  C      +-PROCDB-+-(DBISIZES)
2627  C      |        |
2628  C      |        +-(QAABORT)
2629  C      |        |
2630  C      |        +-(DBITITLE)
2631  C      |        |
2632  C      |        +-(QAABORT)
2633  C      |        |
2634  C      |        +-(DBINVAR)
2635  C      |        |
2636  C      |        +-(QAABORT)
2637  C      |        |
2638  C      |        +-ADDMEM (1)-+-(QAABORT)
2639  C      |        |            |
2640  C      |        |            +-(QAABORT)
2641  C      |        |            |
2642  C      |        |            +-(QAABORT)
2643  C      |        |            |
2644  C      |        |            +-(QAABORT)
2645  C      |        |            |
2646  C      |        |            +-(QAABORT)
2647  C      |        |            |
2648  C      |        |            +-(QAABORT)
2649  C      |        |            |
2650  C      |        |            +-(QAABORT)
2651  C      |        |            |
2652  C      |        |            +-(QAABORT)
2653  C      |        |            |
2654  C      |        |            +-(QAABORT)
2655  C      |        |            |
2656  C      |        |            +-(QAABORT)
2657  C      |        |            |
2658  C      |        |            +-(QAABORT)
2659  C      |        |            |
2660  C      |        |            +-(QAABORT)
2661  C      |        |
2662  C      |        +-DYNMEM (2)-+-MEMADJ (3)-+-(QAABORT)
2663  C      |        |            |             |
2664  C      |        |            |             +-(MDGET)
2665  C      |        |            |             |
2666  C      |        |            |             +-(MDFILL)
2667  C      |        |            |             |
2668  C      |        |            |             +-(MDRSRV)
2669  C      |        |            |             |
2670  C      |        |            |             +-(MDDEBG)
2671  C      |        |            |             |
```

```
C        |        |        |        +-(MDSTAT)
C        |        |        |        |
C        |        |        |        +-(MDLIST)
C        |        |        |        |
C        |        |        |        +-(MDEROR)
C        |        |        |        |
C        |        |        |        +-(QAABORT)
C        |        |        |        |
C        |        |        |        +-(QAABORT)
C        |        |        |        |
C        |        |        |        +-(MDLONG)
C        |        |        |        |
C        |        |        |        +-(MDDEBG)
C        |        |        |        |
C        |        |        |        +-(MDSTAT)
C        |        |        |        |
C        |        |        |        +-(MDLIST)
C        |        |        |        |
C        |        |        |        +-(MDEROR)
C        |        |        |        |
C        |        |        |        +-(QAABORT)
C        |        |        |        |
C        |        |        |        +-(MDCOMP)
C        |        |        |        |
C        |        |        |        +-(MDGIVE)
C        |        |        |        |
C        |        |        |        +-(MDFIND)
C        |        |        |        |
C        |        |        |        +-(MDDEBG)
C        |        |        |        |
C        |        |        |        +-(MDSTAT)
C        |        |        |        |
C        |        |        |        +-(MDLIST)
C        |        |        |        |
C        |        |        |        +-(MDEROR)
C        |        |        |        |
C        |        |        |        +-(QAABORT)
C        |        |        |
C        |        |        +-MEMADJ2>MEMADJ see 3
C        |        |        |
C        |        |        +-MEMADJ3>MEMADJ see 3
C        |        |
C        |        +-ZERODA
C        |        |
C        |        +-DYNMEMC (4)-+-MEMADJC (5)-+-(QAABORT)
C        |        |             |             |
C        |        |             |             +-(MCGET)
C        |        |             |             |
C        |        |             |             +-(MCFILL)
C        |        |             |             |
C        |        |             |             +-(MCRSRV)
C        |        |             |             |
C        |        |             |             +-(MCDEBG)
C        |        |             |             |
C        |        |             |             +-(MCSTAT)
C        |        |             |             |
C        |        |             |             +-(MCLIST)
```

```
2729  C  |  |  |  |
2730  C  |  |  |  +-(MCEROR)
2731  C  |  |  |  |
2732  C  |  |  |  +-(QAABORT)
2733  C  |  |  |  |
2734  C  |  |  |  +-(QAABORT)
2735  C  |  |  |  |
2736  C  |  |  |  +-(MCLONG)
2737  C  |  |  |  |
2738  C  |  |  |  +-(MCFILL)
2739  C  |  |  |  |
2740  C  |  |  |  +-(MCDEBG)
2741  C  |  |  |  |
2742  C  |  |  |  +-(MCSTAT)
2743  C  |  |  |  |
2744  C  |  |  |  +-(MCLIST)
2745  C  |  |  |  |
2746  C  |  |  |  +-(MCEROR)
2747  C  |  |  |  |
2748  C  |  |  |  +-(QAABORT)
2749  C  |  |  |  |
2750  C  |  |  |  +-(MCCOMP)
2751  C  |  |  |  |
2752  C  |  |  |  +-(MCGIVE)
2753  C  |  |  |  |
2754  C  |  |  |  +-(MCFIND)
2755  C  |  |  |  |
2756  C  |  |  |  +-(MCDEBG)
2757  C  |  |  |  |
2758  C  |  |  |  +-(MCSTAT)
2759  C  |  |  |  |
2760  C  |  |  |  +-(MCLIST)
2761  C  |  |  |  |
2762  C  |  |  |  +-(MCEROR)
2763  C  |  |  |  |
2764  C  |  |  |  +-(QAABORT)
2765  C  |  |  |
2766  C  |  |  +-MEMADJC2>MEMADJC see 5
2767  C  |  |  |
2768  C  |  |  +-MEMADJC3>MEMADJC see 5
2769  C  |  |
2770  C  |  +-RECHEA-+-(DBIELBLK)
2771  C  |           |
2772  C  |           +-(QAABORT)
2773  C  |           |
2774  C  |           +-(DBIPRNAM)
2775  C  |           |
2776  C  |           +-(QAABORT)
2777  C  |           |
2778  C  |           +-(DBIPROP)
2779  C  |           |
2780  C  |           +-(QAABORT)
2781  C  |           |
2782  C  |           +-(DBIQAREC)
2783  C  |           |
2784  C  |           +-(QAABORT)
2785  C  |           |
```

```
2786  C     |                  +-(DBINSTEP)
2787  C     |                  |
2788  C     |                  +-(QAABORT)
2789  C     |
2790  C     +-(FILOPEN)
2791  C     |
2792  C     +-(QAABORT)
2793  C     |
2794  C     +-(FILOPEN)
2795  C     |
2796  C     +-(QAABORT)
2797  C     |
2798  C     +-PROGEN-+-SGRDB-+-(FFRDFLDS)
2799  C     |        |       |
2800  C     |        |       +-(QAMAXERR)
2801  C     |        |       |
2802  C     |        |       +-(QAABORT)
2803  C     |        |       |
2804  C     |        |       +-(QAABORT)
2805  C     |        |
2806  C     |        +-ADDMEM1>ADDMEM see 1
2807  C     |        |
2808  C     |        +-DYNMEM see 2
2809  C     |        |
2810  C     |        +-DYNMEMC see 4
2811  C     |        |
2812  C     |        +-ZERODA
2813  C     |        |
2814  C     |        +-RGRDB-+-(FFRDFLDS)
2815  C     |                |
2816  C     |                +-(QAMAXERR)
2817  C     |                |
2818  C     |                +-(QAABORT)
2819  C     |                |
2820  C     |                +-(QAABORT)
2821  C     |
2822  C     +-(FILOPEN)
2823  C     |
2824  C     +-(QAABORT)
2825  C     |
2826  C     +-PROCIN-+-SCHAINS-+-(FFRDFLDS)
2827  C     |        |         |
2828  C     |        |         +-(QAMAXERR)
2829  C     |        |         |
2830  C     |        |         +-(QAMAXERR)
2831  C     |        |         |
2832  C     |        |         +-(QAABORT)
2833  C     |        |
2834  C     |        +-SCATXT-+-(FFRDFLDS)
2835  C     |        |        |
2836  C     |        |        +-(QAMAXERR)
2837  C     |        |        |
2838  C     |        |        +-GETVAL (6)--(QAABORT)
2839  C     |        |        |
2840  C     |        |        +-(QAMAXERR)
2841  C     |        |        |
2842  C     |        |        +-(QAMAXERR)
```

```
C        |         |         |
C        |         |         +-GETVAL see 6
C        |         |         |
C        |         |         +-(QAABORT)
C        |         |
C        |         +-ADDMEM2>ADDMEM see 1
C        |         |
C        |         +-DYNMEM see 2
C        |         |
C        |         +-DYNMEMC see 4
C        |         |
C        |         +-ZERODA
C        |         |
C        |         +-RCHAINS-+-(FFRDFLDS)
C        |         |         |
C        |         |         +-(QAMAXERR)
C        |         |         |
C        |         |         +-(QAMAXERR)
C        |         |         |
C        |         |         +-(QAABORT)
C        |         |         |
C        |         |         +-(QAABORT)
C        |         |         |
C        |         |         +-(QAABORT)
C        |         |         |
C        |         |         +-(QAABORT)
C        |         |
C        |         +-REATXT-+-(FFRDFLDS)
C        |                  |
C        |                  +-(QAMAXERR)
C        |                  |
C        |                  +-(QAMAXERR)
C        |                  |
C        |                  +-GETVAL see 6
C        |                  |
C        |                  +-(QAABORT)
C        |                  |
C        |                  +-(QAABORT)
C        |
C        +-(QABIGLET)
C        |
C        +-PROSDB-+-SCASDB-+-(SDBINIT)
C        |        |         |
C        |        |         +-(QAABORT)
C        |        |         |
C        |        |         +-(SDBCASES)
C        |        |         |
C        |        |         +-(QAABORT)
C        |        |         |
C        |        |         +-(SDBINFO)
C        |        |         |
C        |        |         +-(QAABORT)
C        |        |         |
C        |        |         +-(SDBNINFO)
C        |        |         |
C        |        |         +-(QAABORT)
C        |        |         |
```

```
2900  C     |        |        +-(QAMAXERR)
2901  C     |        |        |
2902  C     |        |        +-(QAMAXERR)
2903  C     |        |        |
2904  C     |        |        +-(QAABORT)
2905  C     |        |        |
2906  C     |        |        +-(QAABORT)
2907  C     |        |        |
2908  C     |        |        +-(QAABORT)
2909  C     |        |        |
2910  C     |        |        +-(QAABORT)
2911  C     |        |        |
2912  C     |        |        +-(QAABORT)
2913  C     |        |        |
2914  C     |        |        +-(QAABORT)
2915  C     |        |        |
2916  C     |        |        +-(QAABORT)
2917  C     |        |
2918  C     |        +-REASDB-+-(STRUPCASE)
2919  C     |        |        |
2920  C     |        |        +-(STRUPCASE)
2921  C     |        |        |
2922  C     |        |        +-(STRUPCASE)
2923  C     |        |        |
2924  C     |        |        +-(SDBRWIND)
2925  C     |        |        |
2926  C     |        |        +-(SDBGETL)
2927  C     |        |        |
2928  C     |        |        +-(QAABORT)
2929  C     |        |        |
2930  C     |        |        +-(STRUPCASE)
2931  C     |        |        |
2932  C     |        |        +-(STRUPCASE)
2933  C     |        |        |
2934  C     |        |        +-(STRUPCASE)
2935  C     |        |        |
2936  C     |        |        +-(STRUPCASE)
2937  C     |        |        |
2938  C     |        |        +-(QAABORT)
2939  C     |        |        |
2940  C     |        |        +-(QAABORT)
2941  C     |        |
2942  C     |        +-GSDBQA-+-(SDBRWIND)
2943  C     |                 |
2944  C     |                 +-(SDBQUERY)
2945  C     |                 |
2946  C     |                 +-(QAABORT)
2947  C     |                 |
2948  C     |                 +-(SDBQUERY)
2949  C     |                 |
2950  C     |                 +-(QAABORT)
2951  C     |                 |
2952  C     |                 +-(SDBRWIND)
2953  C     |                 |
2954  C     |                 +-(EXFILDAT)
2955  C     |
2956  C     +-(DBOOPEN)
```

```
C        |
C        +-(QAABORT)
C        |
C        +-(QAMAKREC)
C        |
C        +-WRCHEA-+-(DBOQAREC)
C        |        |
C        |        +-(QAABORT)
C        |        |
C        |        +-(DBOQAREC)
C        |        |
C        |        +-(QAABORT)
C        |        |
C        |        +-(DBOHEAD)
C        |        |
C        |        +-(QAABORT)
C        |        |
C        |        +-(DBOVRNAM)
C        |        |
C        |        +-(QAABORT)
C        |        |
C        |        +-(DBOVRNAM)
C        |        |
C        |        +-(QAABORT)
C        |        |
C        |        +-(DBOVRNAM)
C        |        |
C        |        +-(QAABORT)
C        |        |
C        |        +-(DBOVRNAM)
C        |        |
C        |        +-(QAABORT)
C        |        |
C        |        +-(DBOVRNAM)
C        |        |
C        |        +-(QAABORT)
C        |
C        +-DECAYW-+-WRTANA (7)-+-(DBOTIME)
C        |        |             |
C        |        |             +-(DBOVAR)
C        |        |             |
C        |        |             +-(QAABORT)
C        |        |             |
C        |        |             +-(DBOVAR)
C        |        |             |
C        |        |             +-(QAABORT)
C        |        |             |
C        |        |             +-(DBOVAR)
C        |        |             |
C        |        |             +-(QAABORT)
C        |        |             |
C        |        |             +-(DBOVAR)
C        |        |             |
C        |        |             +-(QAABORT)
C        |        |             |
C        |        |             +-(DBOVAR)
C        |        |             |
```

```
3014  C       |         |             +-(QAABORT)
3015  C       |         |             |
3016  C       |         |             +-(DBOSTEP)
3017  C       |         |             |
3018  C       |         |             +-(QAABORT)
3019  C       |         |
3020  C       |         +-WRTANA see 7
3021  C       |         |
3022  C       |         +-GINDX (8)--(QAABORT)
3023  C       |         |
3024  C       |         +-GJNDX--(QAABORT)
3025  C       |         |
3026  C       |         +-QUANKG-+-(QAABORT)
3027  C       |         |        |
3028  C       |         |        +-PEXP
3029  C       |         |        |
3030  C       |         |        +-PEXP
3031  C       |         |
3032  C       |         +-WRTANA see 7
3033  C       |         |
3034  C       |         +-WRTANA see 7
3035  C       |
3036  C       +-WRIOUT-+-GINDX see 8
3037  C       |         |
3038  C       |         +-GINDX see 8
3039  C       |         |
3040  C       |         +-WRPLFIL--(QAABORT)
3041  C       |
3042  C       +-WRPTRN
3043  C       |
3044  C       +-(DBPTITLE)
3045  C       |
3046  C       +-(DBPVERS)
3047  C       |
3048  C       +-(DBPQAREC)
3049  C       |
3050  C       +-(DBPSIZES)
3051  C       |
3052  C       +-(DBICLOSE)
3053  C       |
3054  C       +-(QAABORT)
3055  C       |
3056  C       +-(DBOCLOSE)
3057  C       |
3058  C       +-(QAABORT)
3059  C       |
3060  C       +-(QACPUS)
3061  C       |
3062  C       +-(QAPAGE)
3063  C
3064  *COMDECK GEOPAR
3065  C***********************************************************************
3066  C     /GEOPAR/ ($include 'GEOPAR.INC')
3067  C      HEADER = Character description of problem, etc...
3068  C      IMAX   = No. of I-lines in grid
3069  C      JMAX   = No. of J-lines in grid
3070  C      KMAX   = No. of K-lines in grid
```

```
3071  C      LESSEL = List of Element Side sets concatenated Element List
3072  C      LESSNL = List of Element Side sets concatenated Nodes associated
3073  C               with each element side set Node List
3074  C      LNPSNL = List of Nodal Point set concatenated Nodes List
3075  C      MAXNENO= Max. no. of nodes of any element
3076  C      NDIM   = Number of physical DIMensions
3077  C      NDBCAM = CAMDAT versoin number
3078  C      NELBLK = Number of ELement BLocKs in mesh
3079  C      NELX   = No. of elements along X-axis of mesh
3080  C      NELY   = No. of elements along Y-axis of mesh
3081  C      NELZ   = No. of elements along Z-axis of mesh
3082  C      NQAREC = Number of QA records
3083  C      NSTEP  = No. of time steps in input CAMDAT (INTEGER)
3084  C      NUMEL  = NUmber of ELeMents in mesh
3085  C      NUMNOD = NUMber of NODes in mesh
3086  C      NUMNPS = No. of Nodal Point Sets
3087  C      NUMESS = No. of Element Side Sets
3088  C      NUQATR = No. of UniQue ATtRibute names
3089  C      NUQPR0 = No. of UniQUe PRoperty names (old)
3090  C      NUQPRP = No. of UniQUe PRoPerty names (new)
3091  C      NVAREL = No. of ELEMENT type ANALYSIS variables
3092  C      NXINFO = No. of information records
3093  C***********************************************************************
3094  C234567
3095        CHARACTER*(80) HEADER
3096        INTEGER
3097       1IMAX,    JMAX,    KMAX,    MAXNENO, LESSEL,  LESSNL,  LNPSNL,
3098       2NDBCAM,  NDIM,    NELBLK,  NELX,    NELY,    NELZ,    NQAREC,
3099       3NSTEP,   NUMEL,   NUMESS,  NUMNOD,  NUMNPS,  NUQATR,  NUQPR0,
3100       4NUQPRP,  NVARHI,  NVARHI2, NVARGL,  NVARGL2, NVARND,  NVAREL,
3101       5NXINFO
3102        COMMON /GEOPAR/
3103       1IMAX,    JMAX,    KMAX,    MAXNENO, LESSEL,  LESSNL,  LNPSNL,
3104       2NDBCAM,  NDIM,    NELBLK,  NELX,    NELY,    NELZ,    NQAREC,
3105       3NSTEP,   NUMEL,   NUMESS,  NUMNOD,  NUMNPS,  NUQATR,  NUQPR0,
3106       4NUQPRP,  NVARHI,  NVARHI2, NVARGL,  NVARGL2, NVARND,  NVAREL,
3107       5NXINFO
3108  C***********************************************************************
3109
3110  *COMDECK INDEX
3111  C***********************************************************************
3112  C    /INDEX/ ($include 'INDEX.INC')
3113  C     LASCIDX= Last index of CHARACTER*1 dynamic arrays (INTEGER)
3114  C     BCDAIDX= First index of current CHAR*1 dynamic array (INTEGER)
3115  C     ECDAIDX= Last  index of current CHAR*1 dynamic array (INTEGER)
3116  C     LASRIDX= Last index of REAL dynamic arrays (INTEGER)
3117  C     BRDAIDX= First index of current REAL dynamic array (INTEGER)
3118  C     ERDAIDX= Last  index of current REAL dynamic array (INTEGER)
3119  C***********************************************************************
3120  C234567
3121        INTEGER BCDAIDX,BRDAIDX,ECDAIDX,ERDAIDX,LASCIDX,LASRIDX
3122        COMMON /INDEX/ BCDAIDX,BRDAIDX,ECDAIDX,ERDAIDX,LASCIDX,LASRIDX
3123  C***********************************************************************
3124  *COMDECK LOCAT
3125  C***********************************************************************
3126  C    /LOCATC/ ($include 'LOCAT.INC')
3127  C     LCPTR  = Array used for pointing to each dynamic array start
```

```
3128  C                 location in the base CHARACTER*1 array, C.
3129  C      L[name]= Pointer to CHAR dynamic array ( L[dynamic-array-name] )
3130  C*********************************************************************
3131  C     /LOCATR/ ($include 'LOCAT.INC')
3132  C      LRPTR  = Array used for pointing to each dynamic array start
3133  C                 location in the base REAL array, Q.
3134  C      L[name]= Pointer to REAL dynamic array ( L[dynamic-array-name] )
3135  C*********************************************************************
3136  C234567
3137        INTEGER
3138       $LIDEBLK, LNUMPRP, LIASPRP,  LXMATPR, LTIME,    LHIFLAG,
3139       $LISEVOK, LVALGLV, LVALNDV,  LVALELV,  LVALHIV,
3140       $LMAPGEN, LNUMCEL, LRPTR(13)
3141        EQUIVALENCE (LRPTR,LIDEBLK)
3142        COMMON /LOCATR/
3143       $LIDEBLK, LNUMPRP,  LIASPRP,  LXMATPR, LTIME,    LHIFLAG,
3144       $LISEVOK, LVALGLV, LVALNDV,  LVALELV,  LVALHIV,
3145       $LMAPGEN, LNUMCEL
3146        INTEGER
3147       $LQAINFO,  LNAMELB, LNMATPR,
3148       $LSAVNAM, LNAMINF, LCPTR(5)
3149        EQUIVALENCE (LCPTR,LQAINFO)
3150        COMMON /LOCATC/
3151       $LQAINFO,  LNAMELB, LNMATPR,
3152       $LSAVNAM, LNAMINF
3153  C*********************************************************************
3154  *COMDECK MACHDP
3155  C*********************************************************************
3156  C      /MACHDP/ (from FORTRAN statement: INCLUDE 'MACHDP.INC')
3157  C       XLIMDP = MACHDP limit for zero (REAL*8)
3158  C       XMINDP = MACHDP limit lower exponent value (REAL*8)
3159  C       XMAXDP = MACHDP limit upper exponent value (REAL*8)
3160  C       VALMAXD= Largest DOUBLE PRECISION real value allowed (REAL*8)
3161  C*********************************************************************
3162  C234567
3163        REAL*8 VALMAXD, XLIMDP, XMAXDP, XMINDP
3164        COMMON /MACHDP/ VALMAXD, XLIMDP, XMAXDP, XMINDP
3165  C*********************************************************************
3166
3167  *COMDECK MISCDAT
3168  C*********************************************************************
3169  C     /MISCDAT/ (from FORTRAN statement: INCLUDE 'MISCDAT.INC')
3170  C      LTYEARS = Logical variable,
3171  C               =.TRUE.; Time variables used are to be in YEARS
3172  C                (Convert intrusion times specified in SECONDS
3173  C                to YEARS)
3174  C      LTSECON = Logical variable,
3175  C               = .TRUE.; Time variables used are to be in SECONDS
3176  C                 (Convert intrusion times specified in YEARS to
3177  C                  SECONDS)
3178  C      GLOBAL
3179  C*********************************************************************
3180  C234567
3181        INTEGER HIFLAG, NSOLNS
3182        LOGICAL GLOBAL, LTSECON, LTYEARS
3183        COMMON /MISCDAT/ HIFLAG, GLOBAL, LTSECON, LTYEARS, NSOLNS
3184  C*********************************************************************
```

```
3185
3186  *COMDECK PARAMS
3187  C*********************************************************************
3188  C     /PARAMS/ (from FORTRAN statement: INCLUDE 'PARAMS.INC')
3189  C      MAXSOLN = Max. no. of infiltration solutions [INTEGER]
3190  C      MAXCELL  = Max. NUMBER of grid cells where infiltration occurs
3191  [INTEGER]
3192  C*********************************************************************
3193  C234567
3194        INTEGER MAXCELL, MAXSOLN, MAXTIMES
3195        PARAMETER (MAXCELL=25, MAXSOLN=12000, MAXTIMES=2)
3196  C*********************************************************************
3197
3198        INTEGER NUMCELL, NUMSOLN, STEP_STORE
3199        REAL AVE_INF, MAXEAST, MAXNORTH, MINEAST, MINNORTH, TIME_STORE
3200        CHARACTER*8  ARRAY_STORE, NAMINF, NAMNUM, NAMRUN
3201        LOGICAL INCLUDE, SUBSHED
3202        COMMON /POSTINF/ INCLUDE, SUBSHED, NUMCELL, NUMSOLN, STEP_STORE,
3203       +                 AVE_INF, MAXEAST, MAXNORTH,  MINEAST, MINNORTH,
3204       +                 TIME_STORE, ARRAY_STORE, NAMINF, NAMNUM, NAMRUN
3205  *COMDECK QA
3206  C*********************************************************************
3207  C     /QA/ (from FORTRAN statement: INCLUDE 'QA.INC')
3208  C      PRGNAM = Program name (CHAR*12)
3209  C      PRGVER = "       version (CHAR*8)
3210  C      PRGUPD = "       update (CHAR*8)
3211  C      PRGATH = "       author (CHAR*80)
3212  C      PRGSPN = "       sponsor (CHAR*80)
3213  C*********************************************************************
3214  C234567
3215        CHARACTER*(8)  PRGUPD,PRGVER
3216        CHARACTER*(12) PRGNAM
3217        CHARACTER*(80) PRGATH,PRGSPN
3218        COMMON /QA/ PRGNAM,PRGVER,PRGUPD,PRGATH,PRGSPN
3219  C*********************************************************************
3220
3221
3222
```

ATTACHMENT VII

LISTING OF SAMPLE INPUT FOR ROUTINES

Example input files are included for the following routines in order of use in analysis. They appear in this attachment in the order below. Table VII-1 contains a cross reference chart describing the linkage of input files and output files for these routines.

GENMESH
MATSET
PRELHS
LHS
POSTLHS
PREINFIL
INFIL
POSTINFIL
ALGEBRACDB

Table VII-1. Code I/O Files Flowchart Key

Code	Input File	Table No.	Output File	Location of Input Files
GENMESH	*usemame_gn*.inp	VII-2	*usemame_gn*.cdb	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.MATSET]
MATSET	*usemame_gn*.cdb *usemame_ms*.inp	VII-3	*usemame_ms*.cdb	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.MATSET]
PRELHS	Prelhs.inp infil.sdb	VII-4a VII-4b	lhs.inp	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.GLACIAL.PRELHS]
LHS	lhs.inp	VII-5	lhs.out	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.GLACIAL.LHS]
POSTLHS	* *usemame_ms* cdb (from MATSET) postlhs.inp	VII-6	*usemame_lhs_r###*.cdb	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.GLACIAL.LHS]
PREINFIL	*usemamepre*.inp *template*.inp *usemame_lhs_###*.cdb	VII-7,8	*usemame_inf_###*.inp	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.PREINFIL.GLACIAL]
INFIL	*usemame_inf_###*.inp geospatial input climate input	VII-9	infil_###.out (output file containing rate maps)	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.GLACIAL.*watershed_name*]
POSTINFIL	*usemame_ms*.inp	VII-10	*usemame_pl_###*.cdb	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.GLACIAL]
ALGEBRACDB	*usemame_pl*.inp *usemame_pl_###*.cdb	VII-11	*usemame_alg_###*.cdb	I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.GLACIAL]

- CDB – Computational Data Base file
- *usemame* – User defines this name
- .inp suffix denotes input user control file
- ### suffix denotes realizations 1 through 100

Table VII-2. GENMESH Sample Input User Control File for Infiltration Uncertainty Analysis, McCurley (299-1282) December 1999

```
!=================================================================
!
! TITLE:      3-D mesh/grid for Infiltration Uncertainty Analysis
! DATE:       12-JAN-1999
! ANALYST:    R. D. McCurley, SNL/COMPA 299-1282
!
!=================================================================
!
*TITLE
       TITLE=3-D MESH/GRID WITH ELEVATION ELEMENTS (SURFACE NORMAL TO Z)
!
*UNITS
       UNITS=SYSTEM_INTERNATIONAL
!
*SETUP
        DIM=    3
        ORIGIN= 0., 0., 0.
        IJKMAX= 11, 11, 2
!
*GRID
        DEL, COORD=X, DEL=1.0, INRANGE=1,11
        DEL, COORD=Y, DEL=1.0, INRANGE=1,11
        DEL, COORD=Z, DEL=1.0, INRANGE=1,2
!
*REGIONS
! -- test region1 for infiltration
        REGION=1, IRANGE=1,6,  JRANGE=1,11, KRANGE=1,2
        REGION=1, IRANGE=6,11, JRANGE=6,11, KRANGE=1,2
! -- test region1 for infiltration
        REGION=2, IRANGE=6,11, JRANGE=1,6, KRANGE=1,2
!
*ELEVATION_elements, ADJUST_Z_COORDINATES
!
        LOCAT, IRANGE=1,2, JRANGE=1,2, KRANGE=1,2, THICK=1.0, ELEVAT=0.0
!
        LOCAT, IRANGE=2,3, JRANGE=1,2, KRANGE=1,2, THICK=1.5, ELEVAT=0.0
        LOCAT, IRANGE=1,2, JRANGE=2,3, KRANGE=1,2, THICK=1.5, ELEVAT=0.0
!
        LOCAT, IRANGE=3,4, JRANGE=1,2, KRANGE=1,2, THICK=2.0, ELEVAT=0.0
        LOCAT, IRANGE=2,3, JRANGE=2,3, KRANGE=1,2, THICK=2.0, ELEVAT=0.0
        LOCAT, IRANGE=1,2, JRANGE=3,4, KRANGE=1,2, THICK=2.0, ELEVAT=0.0
!
        LOCAT, IRANGE=4,5, JRANGE=1,2, KRANGE=1,2, THICK=2.5, ELEVAT=0.0
        LOCAT, IRANGE=3,4, JRANGE=2,3, KRANGE=1,2, THICK=2.5, ELEVAT=0.0
        LOCAT, IRANGE=2,3, JRANGE=3,4, KRANGE=1,2, THICK=2.5, ELEVAT=0.0
        LOCAT, IRANGE=1,2, JRANGE=4,5, KRANGE=1,2, THICK=2.5, ELEVAT=0.0
!
        LOCAT, IRANGE=5,6, JRANGE=1,2, KRANGE=1,2, THICK=3.0, ELEVAT=0.0
        LOCAT, IRANGE=4,5, JRANGE=2,3, KRANGE=1,2, THICK=3.0, ELEVAT=0.0
        LOCAT, IRANGE=3,4, JRANGE=3,4, KRANGE=1,2, THICK=3.0, ELEVAT=0.0
        LOCAT, IRANGE=2,3, JRANGE=4,5, KRANGE=1,2, THICK=3.0, ELEVAT=0.0
        LOCAT, IRANGE=1,2, JRANGE=5,6, KRANGE=1,2, THICK=3.0, ELEVAT=0.0
!
        LOCAT, IRANGE=6,7, JRANGE=1,2, KRANGE=1,2, THICK=3.5, ELEVAT=0.0
        LOCAT, IRANGE=5,6, JRANGE=2,3, KRANGE=1,2, THICK=3.5, ELEVAT=0.0
        LOCAT, IRANGE=4,5, JRANGE=3,4, KRANGE=1,2, THICK=3.5, ELEVAT=0.0
        LOCAT, IRANGE=3,4, JRANGE=4,5, KRANGE=1,2, THICK=3.5, ELEVAT=0.0
        LOCAT, IRANGE=2,3, JRANGE=5,6, KRANGE=1,2, THICK=3.5, ELEVAT=0.0
        LOCAT, IRANGE=1,2, JRANGE=6,7, KRANGE=1,2, THICK=3.5, ELEVAT=0.0
!
        LOCAT, IRANGE=7,8, JRANGE=1,2, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
        LOCAT, IRANGE=6,7, JRANGE=2,3, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
        LOCAT, IRANGE=5,6, JRANGE=3,4, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
        LOCAT, IRANGE=4,5, JRANGE=4,5, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
        LOCAT, IRANGE=3,4, JRANGE=5,6, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
        LOCAT, IRANGE=2,3, JRANGE=6,7, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
        LOCAT, IRANGE=1,2, JRANGE=7,8, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
```

```
!
      LOCAT, IRANGE=8,9, JRANGE=1,2, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=7,8, JRANGE=2,3, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=6,7, JRANGE=3,4, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=5,6, JRANGE=4,5, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=4,5, JRANGE=5,6, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=3,4, JRANGE=6,7, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=2,3, JRANGE=7,8, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=1,2, JRANGE=8,9, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
!
      LOCAT, IRANGE=9,10, JRANGE=1,2,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=8,9,  JRANGE=2,3,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=7,8,  JRANGE=3,4,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=6,7,  JRANGE=4,5,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=5,6,  JRANGE=5,6,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=4,5,  JRANGE=6,7,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=3,4,  JRANGE=7,8,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=2,3,  JRANGE=8,9,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=1,2,  JRANGE=9,10, KRANGE=1,2, THICK=5.0, ELEVAT=0.0
!!
      LOCAT, IRANGE=10,11, JRANGE=1,2,   KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=9,10,  JRANGE=2,3,   KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=8,9,   JRANGE=3,4,   KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=7,8,   JRANGE=4,5    KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=6,7,   JRANGE=5,6,   KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=5,6,   JRANGE=6,7,   KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=4,5,   JRANGE=7,8,   KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=3,4,   JRANGE=8,9,   KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=2,3,   JRANGE=9,10,  KRANGE=1,2, THICK=5.5, ELEVAT=0.0
      LOCAT, IRANGE=1,2,   JRANGE=10,11, KRANGE=1,2, THICK=5.5, ELEVAT=0.0
!!
      LOCAT, IRANGE=2,3,   JRANGE=10,11, KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=3,4,   JRANGE=9,10,  KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=4,5,   JRANGE=8,9,   KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=5,6,   JRANGE=7,8,   KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=6,7,   JRANGE=6,7,   KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=7,8,   JRANGE=5,6,   KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=8,9,   JRANGE=4,5,   KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=9,10,  JRANGE=3,4,   KRANGE=1,2, THICK=5.0, ELEVAT=0.0
      LOCAT, IRANGE=10,11, JRANGE=2,3,   KRANGE=1,2, THICK=5.0, ELEVAT=0.0
!
      LOCAT, IRANGE=3,4,   JRANGE=10,11, KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=4,5,   JRANGE=9,10,  KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=5,6,   JRANGE=8,9,   KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=6,7,   JRANGE=7,8,   KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=7,8,   JRANGE=6,7,   KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=8,9,   JRANGE=5,6,   KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=9,10,  JRANGE=4,5,   KRANGE=1,2, THICK=4.5, ELEVAT=0.0
      LOCAT, IRANGE=10,11, JRANGE=3,4,   KRANGE=1,2, THICK=4.5, ELEVAT=0.0
!
      LOCAT, IRANGE=4,5,   JRANGE=10,11, KRANGE=1,2, THICK=4.0, ELEVAT=0.0
      LOCAT, IRANGE=5,6,   JRANGE=9,10,  KRANGE=1,2, THICK=4.0, ELEVAT=0.0
      LOCAT, IRANGE=6,7,   JRANGE=8,9,   KRANGE=1,2, THICK=4.0, ELEVAT=0.0
      LOCAT, IRANGE=7,8,   JRANGE=7,8,   KRANGE=1,2, THICK=4.0, ELEVAT=0.0
      LOCAT, IRANGE=8,9,   JRANGE=6,7,   KRANGE=1,2, THICK=4.0, ELEVAT=0.0
      LOCAT, IRANGE=9,10,  JRANGE=5,6,   KRANGE=1,2, THICK=4.0, ELEVAT=0.0
      LOCAT, IRANGE=10,11, JRANGE=4,5,   KRANGE=1,2, THICK=4.0, ELEVAT=0.0
!
      LOCAT, IRANGE=5,6,   JRANGE=10,11, KRANGE=1,2, THICK=3.5, ELEVAT=0.0
      LOCAT, IRANGE=6,7,   JRANGE=9,10,  KRANGE=1,2, THICK=3.5, ELEVAT=0.0
      LOCAT, IRANGE=7,8,   JRANGE=8,9,   KRANGE=1,2, THICK=3.5, ELEVAT=0.0
      LOCAT, IRANGE=8,9,   JRANGE=7,8,   KRANGE=1,2, THICK=3.5, ELEVAT=0.0
      LOCAT, IRANGE=9,10,  JRANGE=6,7,   KRANGE=1,2, THICK=3.5, ELEVAT=0.0
      LOCAT, IRANGE=10,11, JRANGE=5,6,   KRANGE=1,2, THICK=3.5, ELEVAT=0.0
!
      LOCAT, IRANGE=6,7,   JRANGE=10,11, KRANGE=1,2, THICK=3.0, ELEVAT=0.0
      LOCAT, IRANGE=7,8,   JRANGE=9,10,  KRANGE=1,2, THICK=3.0, ELEVAT=0.0
      LOCAT, IRANGE=8,9,   JRANGE=8,9,   KRANGE=1,2, THICK=3.0, ELEVAT=0.0
      LOCAT, IRANGE=9,10,  JRANGE=7,8,   KRANGE=1,2, THICK=3.0, ELEVAT=0.0
      LOCAT, IRANGE=10,11, JRANGE=6,7,   KRANGE=1,2, THICK=3.0, ELEVAT=0.0
!
```

```
        LOCAT, IRANGE=7,8,   JRANGE=10,11, KRANGE=1,2, THICK=2.5, ELEVAT=0.0
        LOCAT, IRANGE=8,9,   JRANGE=9,10,  KRANGE=1,2, THICK=2.5, ELEVAT=0.0
        LOCAT, IRANGE=9,10,  JRANGE=8,9,   KRANGE=1,2, THICK=2.5, ELEVAT=0.0
        LOCAT, IRANGE=10,11, JRANGE=7,8,   KRANGE=1,2, THICK=2.5, ELEVAT=0.0
!
        LOCAT, IRANGE=8,9,   JRANGE=10,11, KRANGE=1,2, THICK=2.0, ELEVAT=0.0
        LOCAT, IRANGE=9,10,  JRANGE=9,10,  KRANGE=1,2, THICK=2.0, ELEVAT=0.0
        LOCAT, IRANGE=10,11, JRANGE=8,9,   KRANGE=1,2, THICK=2.0, ELEVAT=0.0
!
        LOCAT, IRANGE=9,10,  JRANGE=10,11, KRANGE=1,2, THICK=1.5, ELEVAT=0.0
        LOCAT, IRANGE=10,11, JRANGE=9,10,  KRANGE=1,2, THICK=1.5, ELEVAT=0.0
!
        LOCAT, IRANGE=10,11, JRANGE=10,11, KRANGE=1,2, THICK=1.0, ELEVAT=0.0
!
!*BOUNDARY_FLAGS
!
*END
```

Table VII-3. Sample Input User Control File for MATSET 9.01 routine, McCurley (299-1282) December 1999

```
!=================================================================
!
!   TITLE:    MATSET input text file for MATSET 9.01
!   ANALYST:  Ron McCurley - SNL/COMPA
!   DATE:     May-1999
!
!=================================================================
!
*PRINt_assigned_values
!
*HEADING
       SCENario, Test_YMP
       SCALe,    LOCAL
       TITLe,    MATSET 9.01: CAMDAT_LIB,SDBREAD_LIB & RENAMING loaded
!
*UNITS="System Internationale"
!
!*CREATE_blocks
!       BLOCK_IDS=1,2
!
*RETRieve*name
       COORDinates, DIM=3, NAMES=X,Y,Z
       MATerials, 1=FOOTPRNT,         2=OUTSIDE
!
       PROPerties, MATERIAL=FOOTPRNT,  NAMES= BRPOROS, BRZDEPTH, SOILDEPM, &
                                              PRECIPM, POTETMUL, BRPERM,  &
                                              SOILPERM, ETCOEFFA, ETCOEFFB, &
                                              FLAREA, ATEMP1, SNOPAR1,SUBPAR1
       PROPerties, MATERIAL=OUTSIDE, NAMES= BRPOROS, BRZDEPTH, SOILDEPM, &
                                            PRECIPM, POTETMUL, BRPERM,  &
                                            SOILPERM, ETCOEFFA, ETCOEFFB, &
                                            FLAREA, ATEMP1, SNOPAR1, SUBPAR1
!
*SET_values
!
       PROPerty_values, MATERIAL=FOOTPRNT,NAME*values: BRPOROS=.02, &
                                          BRZDEPTH=1.5, &
                                          SOILDEPM=1.0, PRECIPM=1.0, &
                                          POTETMUL=1.0, BRPERM=1.0,  &
                                          SOILPERM=1.0, ETCOEFFA=-1.5, &
                                          ETCOEFFB=1.26, SNOPAR1=1.78, &
                                          SUBPAR1 = 0.1, FLAREA=0.5, &
                                          ATEMP1=17.3
       PROPerty_values, MATERIAL=OUTSIDE,NAME*values: BRPOROS=.02, &
                                          BRZDEPTH=1.5, &
                                          SOILDEPM=1.0, PRECIPM=1.0, &
                                          POTETMUL=1.0, BRPERM=1.0,  &
                                          SOILPERM=1.0, ETCOEFFA=-1.5, &
                                          ETCOEFFB=1.26, SNOPAR1=1.78, &
                                          SUBPAR1 = 0.1, FLAREA=0.5, &
                                          ATEMP1=17.3
  !=================================================================
*END
```

Table VII-4a. Sample Input User Control File for PRELHS routine, McCurley (299-1282) December 1999

```
!=====================================================================
!  TITLE:     PRELHS input file == PRELHS.INP
!  ANALYST:   Ron McCurley, Aug 19, 1999  COMPA/SNL 505-299-1282
!=====================================================================
!
!  DESCRIPTION:
!  PRELHS Variable Sampling Input File for Glacial Transition Climate
!
!========= No Comments Allowed between *ECHO and *ENDECHO ==============
*ECHOLHS
TITLE BEGIN MONTE CARLO, Nov 8, 1999 SAMPLING FOR FOOTPRINT SET WATERSHEDS
NOBS                100
RANDOM SEED         0733050198
OUTPUT CORR HIST DATA PLOT
*ENDECHO
!
!==== PROPERTIES TO BE RETRIEVED FROM YMP DATABASE, '.SDB' ====
*RETRIEVE
!
   MATERIALS,  FOOTPRNT
   PROPERTIES, BRPOROS, BRZDEPTH, SOILDEPM, PRECIPM, POTETMUL, BRPERM, &
               SOILPERM, ETCOEFFA, ETCOEFFB, FLAREA, SUBPAR1, SNOPAR1
!
!=====================================================================
!
*END
```

Table VII-4b. ASCII Datafile Containing Uncertain Parameter Distributions Used to Set Up Input Files for LHS, McCurley (299-1282) December 1999

```
*HEADERLINES=1
*ENTRY=IDMTRL,IDPRAM
ID          2:5        I4         'ID'
IDMTRL      7:14       A8         'MATERIAL ID'
IDPRAM     16:23       A8         'PARAMETER ID'
MEAN       24:35       1PE12.4      'MEAN'
MEDIAN     36:47       1PE12.4      'MEDIAN'
LOWRNG     48:59       1PE12.4      'MINIMUM'
HIRNGE     60:71       1PE12.4      'MAXIMUM'
VRYFRC     72:83       1PE12.4      'COEFFICIENT OF VARIATION'
SCALE      84:95       1PE12.4      'PROBABILITY'
DISTYP     97:108      A12        'DISTRIBUTION'
UNITS     110:121      A12        'UNITS'
IDWELL    123:130      A8         'DUMMY WELL ID'
IDSCAL    123:130      A8         'DUMMY SCALE ID'
*NMGEOL=INFIL.SDB
*NSDBVR=1
*ENDDSCR
, BRZDEPTH, SOILDEPM, PRECIPM, POTETMUL, BRPERM,
                SOILPERM, ETCOEFFA
```

ID	IDMTRL	IDPRAM	MEAN	MEDIAN	LOWRNG	HIRNGE	VRYFRC	SCALE	DISTYP	UNITS
1001	FOOTPRNT	BRPOROS	2.0000E-02	2.0000E-02	0.0000E+00	4.0000E-02	0.0000E+00	0.0000E+00	NORMAL	NONE
1002	FOOTPRNT	BRZDEPTH	3.0000E+00	3.0000E+00	1.0000E+00	5.0000E+00	0.0000E+00	0.0000E+00	NORMAL	m
1003	FOOTPRNT	SOILDEPM	1.0000E+00	1.0000E+00	5.0000E-01	1.5000E+00	0.0000E+00	0.0000E+00	NORMAL	NONE
1004	FOOTPRNT	PRECIPM	1.0000E+00	1.0000E+00	6.0000E-01	1.4000E+00	0.0000E+00	0.0000E+00	NORMAL	NONE
1005	FOOTPRNT	POTETMUL	1.0000E+00	1.0000E+00	6.0000E-01	1.4000E+00	0.0000E+00	0.0000E+00	NORMAL	NONE
1006	FOOTPRNT	BRPERM	1.0000E+00	1.0000E+00	5.0000E-02	2.0000E+01	0.0000E+00	0.0000E+00	LOGNORMAL	NONE
1007	FOOTPRNT	SOILPERM	1.0000E+00	1.0000E+00	5.0000E-02	2.0000E+01	0.0000E+00	0.0000E+00	LOGNORMAL	NONE
1008	FOOTPRNT	ETCOEFFA	1.0000E+01	1.0000E+01	5.0000E+00	1.5000E+01	0.0000E+00	0.0000E+00	NORMAL	NONE
1009	FOOTPRNT	ETCOEFFB	1.0400E+00	1.0400E+00	5.4000E-01	1.5400E+00	0.0000E+00	0.0000E+00	NORMAL	NONE
1010	FOOTPRNT	FLAREA	2.5000E-01	2.5000E-01	1.0000E-02	4.9000E-01	0.0000E+00	0.0000E+00	NORMAL	NONE
1011	FOOTPRNT	SNOPAR1	1.7800E+00	1.7800E+00	7.8000E-01	2.7800E+00	0.0000E+00	0.0000E+00	UNIFORM	NONE
1012	FOOTPRNT	SUBPAR1	1.0000E-01	1.0000E-01	0.0000E+00	2.0000E-01	0.0000E+00	0.0000E+00	UNIFORM	NONE
1013	FOOTPRNT	ATEMP1	1.7300E+01	1.7300E+01	1.2300E+01	2.2300E+01	0.0000E-02	0.0000E+00	NORMAL	NONE

Table VII-5. Input file for LHS that is automatically produced by PRELHS routine.

```
TITLE SDB Name = INFIL.SDB        , Ver = 1                12/01/99 17:58:05
TITLE BEGIN MONTE CARLO, Nov 8, 1999 SAMPLING FOR FOOTPRINT SET WATERSHEDS
NOBS                 100
RANDOM SEED          0733050198
NORMAL                   FOOTPRNT  BRPOROS
    0.00000E+00    4.00000E-02
NORMAL                   FOOTPRNT  BRZDEPTH
    1.00000E+00    5.00000E+00
NORMAL                   FOOTPRNT  SOILDEPM
    5.00000E-01    1.50000E+00
NORMAL                   FOOTPRNT  PRECIPM
    6.00000E-01    1.40000E+00
NORMAL                   FOOTPRNT  POTETMUL
    6.00000E-01    1.40000E+00
LOGNORMAL                FOOTPRNT  BRPERM
    5.00000E-02    2.00000E+01
LOGNORMAL                FOOTPRNT  SOILPERM
    5.00000E-02    2.00000E+01
NORMAL                   FOOTPRNT  ETCOEFFA
    5.00000E+00    1.50000E+01
NORMAL                   FOOTPRNT  ETCOEFFB
    5.40000E-01    1.54000E+00
NORMAL                   FOOTPRNT  FLAREA
    1.00000E-02    4.90000E-01
UNIFORM                  FOOTPRNT  SUBPAR1
    0.00000E+00    2.00000E-01
UNIFORM                  FOOTPRNT  SNOPAR1
    7.80000E-01    2.78000E+00
OUTPUT CORR HIST DATA PLOT
TITLE SDB Name = INFIL.SDB        , Ver = 1                12/01/99 17:58:05
```

Table VII-6. Sample Input User Control File for POSTLHS Routine which Post-processes the LHS Output, McCurley (299-1282) December 1999

```
!=====================================================================
!  TITLE:    POSTLHS input file == POSTLHS.INP
!  ANALYST:  R. McCurley--  10/14/99 SNL/COMPA 505-299-1282
!=====================================================================
!
!  DESCRIPTION:  YMP INFIL FOOTPRINT WATERSHEDS ANALYSIS, Glacial
!                Transition
!
!=====================================================================
*ATTRIBUTE_STORE
   MATERIALS  = FOOTPRNT , &
!   ATTRIBUTE_NAMES =
   PROPERTY_NAMES = BRPOROS, BRZDEPTH, SOILDEPM, PRECIPM, POTETMUL, BRPERM, &
                    SOILPERM, ETCOEFFA, ETCOEFFB, FLAREA, SUBPAR1, SNOPAR1
!
!=====================================================================
*END
```

Table VII-7. Sample Input User Control File for PREINFIL Routine, McCurley (299-1282) December 1999

```
!PREINFIL user input file -- alpha version
!Analyst--Ron McCurley SNL/COMPA  299-1282
! Date Oct 21, 1999
!Begin input
*FILES
 NAMES: MAIN=INFILMN_ww1_R###.OUT, ECHO=INFILECHO_R###.OUT, &
        SUMMARY=INFILSM_ww1_R###.OUT
*REPLACE
   REGION: FOOTPRNT
   PROPERTIES: PPTFACT=PRECIPM,ETFACT=POTETMUL,RDEPTH3=BRZDEPTH, &
       SKSFACT=SOILPERM, FLAREA=SAME, &
       SDFACT=SOILDEPM, RKPOR=BRPOROS, IMBFACT=BRPERM, BARSOIL1=ETCOEFFA, &
       BARSOIL2=ETCOEFFB, SUBPAR1=SAME, SNOPAR1=SAME
*END
```

Table VII-8. Template Input User Control File for INFIL Version A_2a.1 Input to PREINFIL

(DTN: GS00308311221.011)

```
INFIL2a1.ctl: Solitario East, Tule Lake, ww1-2a1-gm1-k2-w20 (11/18/1999)
1   0.00000001                                  IROUT(1 = coupled, 0 = uncoupled, -1 = flow routing off, -2 = infil off), IFRTOL
2   2     1.78                                      ISNOW, ISNWMOD, SNOPAR1
3   0.1   0.3                                   ISUBLIM, SUBPAR1, SUBPAR2
0   1.                                          IPPTTEST (1 for testing), PPTTEST = constant
0   .0                                          IETTEST (1 for testing), ETTEST = constant
30.0                                                CELSIZE (node spacing (meters): using for flow volume calculations)
544721.0  4072203.0                             xcfs,ycfs: coords for discharge cell
1950  1    1996    366   1   5                 yr1 = start year, dnn1 = start day, yr2 = end year, dn2 = end day
1     1    1    1                              multipliers ( pptfact, etfact, imbfact, sksfact)
1.0   0    0.4                                 SDFACT (soil depth multiplier), IVEGC (set to 1 for map data), FVEGC (use if IVEGC =
0)
1.0   0.8   0.8    0.2                          ROOTF1,ROOTF2,ROOTF3,ROOTF4
1.0   1.0   0.5    0.1                          MAXWGT1, MAXWGT2, MAXWGT3, MAXWGT4
0.3   2.0   3.0    2.0                           RDEPTH1,RDEPTH2,RDEPTH3,RDEPTH4,RDEPTHF
0.02  1.0  .25                                  RKPOR, RKMMFACT, FLAREA
1   1    0                                      INFMOD, ETMOD, RUNMOD
 -10.0  1.04                                 BARSOIL1, BARSOIL2:  bare soil et parameters
3   17.3  11.74                                 IAIRTEMP = 1 for new air temp model, ATEMP1 = avg. air temp, ATEMP2 = air temp
seasonal deviation
2                                               HSTEP: time step for PET model (hours)
5  181  3                                       PPTYUC (=5 diminished elev. correlation, =2 for 4JA , =1 for simple elevation
transfer), AAPREPX, IPPTDAT
Tulelake.inp                                    input file name:  daily precip
     0      1    0.5      1     1               dpthflag, irtz, delvwcf, moistcr, fracmod
     0      1.75                                IVWCFLG,   vwcfact
ww1.w20                                         input file name:  map parameters (*.inp)
-1
ww1-gm1.2a1                                       dayall: average daily mass balance terms
1    1                                       ndaymap, imap
70   1995
ww1-gm1.2a2
ww1-gm1.2a3                                       output file name:  daily mass balance
ww1-gm1.2a4                                       output file name:  total daily fluxes
1      1                                       dbgflag, dbgflag2
ww1-gm1.2a5                                       output file name:  annual mass balance terms
0                                               IDEBUG:  debugging option parameter
ww1-gm1.2a6                                       new debug file
ww1-gm1                                         map output:  annual totals or mult-year averages
--
--------------- Parameters for dynamic root-zone function -------------------------------------
--     depth  rtza   rtzb   rtzc  rtzd  bsoil    delvwc
--     m
     4
```

```
    1    0.5     15      3     3     2     0.30      0.50
    2    1.5     15      4     2     2     0.20      0.50
    3    4.5     10    1.5     1     2     0.10      0.50
    4    6.0     10    1.5     1     2     0.05      0.25
--

--------------- Soil Properties (Brooks & Corey/van Genuchten combined) ----------------------------------------
--     fdcp   etresidpor    beta  alpah ksat     PE        B     n     vg-alpha sorp     SOILP   potis
--                                     Kg s /m^3J/Kg                   1/(J/Kg) ???              J/Kg
   10
    1  0.242  0.054  0.366  -3.5  1.26  5.6E-04 -1.19E+00  4.72  1.24  5.2E-01    0.390    0.05 -1.0E+02
    2  0.173  0.023  0.315  -3.5  1.26  1.2E-03 -9.41E-01  3.70  1.31  6.2E-01    0.500    0.05 -1.0E+02
    3  0.163  0.017  0.325  -3.5  1.26  1.3E-03 -8.60E-01  3.36  1.36  6.6E-01    0.510    0.05 -1.0E+02
    4  0.073  0.002  0.281  -3.5  1.26  3.8E-03 -6.22E-01  2.18  1.62  8.7E-01    0.700    0.05 -1.0E+02
    5  0.200  0.035  0.330  -3.5  1.26  6.7E-04 -1.07E+00  4.14  1.78  5.6E-01    0.400    0.05 -1.0E+02
    6  0.150  0.011  0.339  -3.5  1.26  2.7E-03 -7.55E-01  3.06  1.40  7.4E-01    0.700    0.05 -1.0E+02
    7  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
    8  0.234  0.046  0.370  -3.5  1.26  5.6E-04 -1.10E+00  4.43  1.26  5.5E-01    0.390    0.05 -1.0E+02
    9  0.189  0.028  0.322  -3.5  1.26  5.7E-04 -1.08E+00  3.88  1.30  5.5E-01    0.370    0.05 -1.0E+02
   10  0.189  0.028  0.322  -3.5  1.26  5.7E-05 -1.08E+00  3.88  1.00  5.5E-01    0.037    0.05 -1.0E+02
--
---------------Rock1 Properties (Brooks & Corey/van Genuchten combined)----------------------------------------
--     rkcp   etresidpor    beta  alpha ksat     PE        B     n     vg-alpha fracks   imbibe  potir
--                                     Kg s /m^3J/Kg                   1/(J/Kg) Kg s /m^3mm/dy   J/Kg
   129
    1    0.35    0.065    0.366    -1.5    1.26   0.0011    -0.93     3.6    1.25    0.52  0.00056     500    -100
    2   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.41    -100
    3   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
    4   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
    5   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
    6   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.41    -100
    7   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
    8   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
    9   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
   10   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
   11   0.105     0.02     0.11    -1.5    1.26  4.0E-09       -6    3.92    1.47      64  2.4E-07    0.06    -100
   12   0.388    0.039    0.393    -1.5    1.26 0.000053     -242    4.57    1.38    2400 0.000053    2.74    -100
   13    0.43    0.044    0.435    -1.5    1.26   0.0004    -1790    6.81    1.23   18000   0.0004   13.83    -100
   14   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
   15   0.401     0.07    0.406    -1.5    1.26 0.000052    -4150    6.78    1.23   42000 0.000052   46.66    -100
   16   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.09    -100
   17   0.043    0.005    0.048    -1.5    1.26  1.6E-07    -88.1    6.43    1.25     890  3.7E-07    0.35    -100
   18   0.248     0.01    0.253    -1.5    1.26  3.8E-06    -82.3    2.71    1.84     830  3.9E-06    3.34    -100
   19   0.159     0.01    0.164    -1.5    1.26  1.2E-06     -140    3.61    1.53    1400  1.3E-06    1.13    -100
   20   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
   21   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
   22   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
   23   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
   24   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
   25   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
   26   0.077    0.011    0.082    -1.5    1.26  5.4E-09      -12    3.04    1.69     120  1.4E-07    0.06    -100
```

27	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
28	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
29	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
30	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
31	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
32	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
33	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
34	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
35	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
36	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
37	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
38	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
39	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
40	0.494	0.05	0.499	-1.5	1.26	0.000088	-341	4.19	1.43	3400	0.000088	75.62	-100
41	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
42	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
43	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
44	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-26.9	5.6	1.29	270	2.1E-07	0.09	-100
45	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
99	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
46	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
47	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
48	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
49	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
50	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
51	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
52	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
53	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
54	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
55	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
56	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
57	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
58	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
59	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
60	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
61	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
62	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
63	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
64	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
65	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
66	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
67	0.031	0.018	0.036	-1.5	1.26	5.0E-09	-0.6	3.39	1.58	10	2.4E-07	0.17	-100
68	0.168	0.021	0.173	-1.5	1.26	7.3E-08	-125	5.36	1.31	1300	1.3E-07	0.07	-100
69	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.02	-100
70	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
71	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.01	-100
72	0.317	0.023	0.322	-1.5	1.26	3.8E-08	-181	4.01	1.46	1800	9.9E-08	1.65	-100
73	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
74	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
75	0.281	0.051	0.286	-1.5	1.26	1.7E-08	-17.5	4.01	1.45	180	7.8E-08	0.02	-100
76	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100

77	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
201	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
202	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.06	-100
203	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
204	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	120	1.4E-07	500	-100
205	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
206	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
207	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
208	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
209	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
210	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
211	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
212	0	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
213	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
214	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
301	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
302	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
303	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
304	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
305	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
306	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
307	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
308	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
309	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
310	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
311	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
312	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
313	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
314	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
315	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
316	0.159	0.01	0.164	-1.5	1.26	1.2E-06	-140	3.61	1.53	1400	1.3E-06	1.13	-100
317	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
318	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
319	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
320	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
321	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
322	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
323	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
324	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
325	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
326	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
327	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
328	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
329	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
330	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
331	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
332	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
333	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
334	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
335	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
336	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100

345	0.097	0.01	0.102	-1.5	-1.26	1.3E-07	-154.6	5.02	1.56	1560	2.9E-07	0.24	-100

Table VII-9. Sample Input User Control File for INFIL Version A_2a.1 Produced by PREINFIL, McCurley (299-1282) December 1999

```
INFIL2a1.ctl: Solitario East, Tule Lake, ww1-2a1-gm1-k2-w20 (11/18/1999)
1   0.00000001                                  IROUT(1 = coupled, 0 = uncoupled, -1 = flow routing off, -2 = infil off), IFRTOL
   2   2 2.16E+00                               ISNOW   ISNWMOD SNOPAR1
  3 1.57E-01 3.00E-01                           ISUBLIM SUBPAR1 SUBPAR2
0   1.                                          IPPTTEST (1 for testing), PPTTEST = constant
0   .0                                          IETTEST (1 for testing), ETTEST = constant
30.0                                                CELSIZE (node spacing (meters): using for flow volume calculations)
544721.0  4072203.0                             xcfs,ycfs: coords for discharge cell
1950  1    1996    366   1   5                 yr1 = start year, dnn1 = start day, yr2 = end year, dn2 = end day
1.12E+00 9.01E-01 5.48E+00 2.23E+00             MULTIPLI(       PPTFACT ETFACT  IMBFACT SKSFACT)
1.13E+00   0  4.00E-01                          SDFACT  (SOIL   DEPTH
1.0   0.8   0.8    0.2                         ROOTF1,ROOTF2,ROOTF3,ROOTF4
1.0   1.0   0.5    0.1                         MAXWGT1, MAXWGT2, MAXWGT3, MAXWGT4
  0.30  2.00  3.75  2.00                          RDEPTH1 RDEPTH2 RDEPTH3 RDEPTH4
2.95E-02  1.00E+00  3.04E-01                       RKPOR    RKMMFACT FLAREA
1   1    0                                      INFMOD, ETMOD, RUNMOD
 -10.610   1.102                                BARSOIL1 BARSOIL2
3   17.3  11.74                                 IAIRTEMP = 1 for new air temp model, ATEMP1 = avg. air temp, ATEMP2 = air temp seaso
2                                               HSTEP: time step for PET model (hours)
5  181  3                                       PPTYUC (=5 diminished elev. correlation, =2 for 4JA , =1 for simple elevation transf
Tulelake.inp                                    input file name:  daily precip
     0      1    0.5      1     1               dpthflag, irtz, delvwcf, moistcr, fracmod
     0      1.75                                 IVWCFLG,   vwcfact
ww1.w20                                         input file name:  map parameters (*.inp)
-1
ww1-gm1.2a1                                      dayall: average daily mass balance terms
1    1                                       ndaymap, imap
70   1995
ww1-gm1.2a2
INFILSM_WW1_R001.OUT
INFILMN_WW1_R001.OUT
1      1                                         dbgflag, dbgflag2
ww1-gm1.2a5                                       output file name:  annual mass balance terms
0                                                 IDEBUG:  debugging option parameter
INFILECHO_R001.OUT
ww1-gm1                                           map output:  annual totals or mult-year averages
--
--------------- Parameters for dynamic root-zone function -------------------------------------
--     depth  rtza   rtzb   rtzc  rtzd  bsoil    delvwc
--     m
     4
     1    0.5     15      3     3     2     0.30      0.50
     2    1.5     15      4     2     2     0.20      0.50
     3    4.5     10    1.5     1     2     0.10      0.50
     4    6.0     10    1.5     1     2     0.05      0.25
--
```

---------- Soil Properties (Brooks & Corey/van Genuchten combined) ----------

--	fdcp	etresidpor	beta	alpah	ksat	PE	B	n	vg-alpha	sorp	SOILP	potis
--					Kg s /m^3	J/Kg			1/(J/Kg)	???		J/Kg
10												
1	0.242	0.054	0.366	-3.5	1.26	5.6E-04	-1.19E+00	4.72	1.24	5.2E-01	0.390	0.05 -1.0E+02
2	0.173	0.023	0.315	-3.5	1.26	1.2E-03	-9.41E-01	3.70	1.31	6.2E-01	0.500	0.05 -1.0E+02
3	0.163	0.017	0.325	-3.5	1.26	1.3E-03	-8.60E-01	3.36	1.36	6.6E-01	0.510	0.05 -1.0E+02
4	0.073	0.002	0.281	-3.5	1.26	3.8E-03	-6.22E-01	2.18	1.62	8.7E-01	0.700	0.05 -1.0E+02
5	0.200	0.035	0.330	-3.5	1.26	6.7E-04	-1.07E+00	4.14	1.78	5.6E-01	0.400	0.05 -1.0E+02
6	0.150	0.011	0.339	-3.5	1.26	2.7E-03	-7.55E-01	3.06	1.40	7.4E-01	0.700	0.05 -1.0E+02
7	0.234	0.046	0.370	-3.5	1.26	5.6E-04	-1.10E+00	4.43	1.26	5.5E-01	0.390	0.05 -1.0E+02
8	0.234	0.046	0.370	-3.5	1.26	5.6E-04	-1.10E+00	4.43	1.26	5.5E-01	0.390	0.05 -1.0E+02
9	0.189	0.028	0.322	-3.5	1.26	5.7E-04	-1.08E+00	3.88	1.30	5.5E-01	0.370	0.05 -1.0E+02
10	0.189	0.028	0.322	-3.5	1.26	5.7E-05	-1.08E+00	3.88	1.00	5.5E-01	0.037	0.05 -1.0E+02

--

----------Rock1 Properties (Brooks & Corey/van Genuchten combined)----------

--	rkcp	etresidpor	beta	alpha	ksat	PE	B	n	vg-alpha	fracks	imbibe	potir
--					Kg s /m^3	J/Kg			1/(J/Kg)	Kg s /m^3	mm/dy	J/Kg
129												
1	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500 -100
2	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41 -100
3	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66 -100
4	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09 -100
5	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66 -100
6	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41 -100
7	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66 -100
8	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09 -100
9	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66 -100
10	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09 -100
11	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.06 -100
12	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74 -100
13	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83 -100
14	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09 -100
15	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66 -100
16	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09 -100
17	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35 -100
18	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34 -100
19	0.159	0.01	0.164	-1.5	1.26	1.2E-06	-140	3.61	1.53	1400	1.3E-06	1.13 -100
20	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
21	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
22	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
23	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
24	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
25	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
26	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
27	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
28	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
29	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100
30	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06 -100

31	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
32	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
33	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
34	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
35	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
36	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
37	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
38	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
39	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
40	0.494	0.05	0.499	-1.5	1.26	0.000088	-341	4.19	1.43	3400	0.000088	75.62	-100
41	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
42	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
43	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
44	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-26.9	5.6	1.29	270	2.1E-07	0.09	-100
45	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
99	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
46	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
47	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
48	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
49	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
50	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
51	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
52	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
53	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
54	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
55	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
56	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
57	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
58	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
59	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
60	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
61	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
62	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
63	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
64	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
65	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.07	-100
66	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
67	0.031	0.018	0.036	-1.5	1.26	5.0E-09	-0.6	3.39	1.58	10	2.4E-07	0.17	-100
68	0.168	0.021	0.173	-1.5	1.26	7.3E-08	-125	5.36	1.31	1300	1.3E-07	0.07	-100
69	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.02	-100
70	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
71	0.327	0.066	0.332	-1.5	1.26	4.5E-09	-39	5.66	1.29	390	6.6E-08	0.01	-100
72	0.317	0.023	0.322	-1.5	1.26	3.8E-08	-181	4.01	1.46	1800	9.9E-08	1.65	-100
73	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
74	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
75	0.281	0.051	0.286	-1.5	1.26	1.7E-08	-17.5	4.01	1.45	180	7.8E-08	0.02	-100
76	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
77	0.043	0.002	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.25	-100
201	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
202	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.06	-100
203	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100

204	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	120	1.4E-07	500	-100
205	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
206	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.41	-100
207	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
208	0.089	0.03	0.094	-1.5	1.26	7.5E-09	-1.8	2.31	2.14	22	2.5E-07	0.14	-100
209	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
210	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.09	-100
211	0.112	0.011	0.117	-1.5	1.26	4.1E-08	-3.2	3.07	1.68	36	1.0E-07	0.09	-100
212	0	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
213	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
214	0.232	0.024	0.237	-1.5	1.26	3.8E-07	-6.8	3.31	1.6	72	4.4E-07	0.05	-100
301	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
302	0.35	0.065	0.366	-1.5	1.26	0.0011	-0.93	3.6	1.25	0.52	0.00056	500	-100
303	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
304	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
305	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
306	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
307	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
308	0.401	0.07	0.406	-1.5	1.26	0.000052	-4150	6.78	1.23	42000	0.000052	46.66	-100
309	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
310	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
311	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
312	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
313	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
314	0.248	0.01	0.253	-1.5	1.26	3.8E-06	-82.3	2.71	1.84	830	3.9E-06	3.34	-100
315	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
316	0.159	0.01	0.164	-1.5	1.26	1.2E-06	-140	3.61	1.53	1400	1.3E-06	1.13	-100
317	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
318	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
319	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
320	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
321	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
322	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
323	0.388	0.039	0.393	-1.5	1.26	0.000053	-242	4.57	1.38	2400	0.000053	2.74	-100
324	0.43	0.044	0.435	-1.5	1.26	0.0004	-1790	6.81	1.23	18000	0.0004	13.83	-100
325	0.494	0.05	0.499	-1.5	1.26	0.000088	-4000	3.78	1.49	40000	0.000088	75.62	-100
326	0.077	0.011	0.082	-1.5	1.26	5.4E-09	-12	3.04	1.69	120	1.4E-07	0.06	-100
327	0.485	0.05	0.49	-1.5	1.26	0.00042	-5260	5.86	1.28	53000	0.00042	276.49	-100
328	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
329	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
330	0.043	0.005	0.048	-1.5	1.26	1.6E-07	-88.1	6.43	1.25	890	3.7E-07	0.35	-100
331	0.151	0.007	0.156	-1.5	1.26	1.7E-07	-377	2.11	2.4	3800	2.8E-07	0.20	-100
332	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
333	0.149	0.01	0.154	-1.5	1.26	2.0E-08	-65.3	5.09	1.33	660	1.2E-07	0.05	-100
334	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
335	0.105	0.02	0.11	-1.5	1.26	4.0E-09	-6	3.92	1.47	64	2.4E-07	0.09	-100
336	0.125	0.01	0.13	-1.5	1.26	2.3E-08	-26.9	5.6	1.29	270	2.2E-07	0.09	-100
345	0.097	0.01	0.102	-1.5	1.26	1.3E-07	-154.6	5.02	1.56	1560	2.9E-07	0.24	-100

Table VII-10. Sample Input User Control File for POSTINFIL Routine, McCurley (299-1282) December 1999

```
!=========================================================================
!  POSTINFIL input file
!  ANALYST:  Ron McCurley, Oct 22, 1999
!
!  DESCRIPTION:
!   POSTINFIL Input File for ww1 watershed, glacial transition climate
!
!=========================================================================
!Store mean infiltration and no. of active cells for watershed WW1
*STORE, &
 ARRAY=GLOBAL, STEP=1, COPY=0, &
 NAMES: MEAN_INFIL=INF_WW1,NUMBER_CELLS=NCELLWW1,MEAN_RUNOFF=RUN_WW1
```

Table VII-11. Sample Input File for ALGEBRACDB for Obtaining Weighted Average over Repository Footprint, Glacial Transition Climate, McCurley (299-1282) December 1999

```
!=====================================================================
!  ALGEBRACDB input file
!  ANALYST:  Ron McCurley, Oct 22, 1999
!
!  DESCRIPTION:
!    ALGEBRACDB Input File for obtaining weighted average over repository
!    footprint, glacial transition climate
!
!=====================================================================
!$ compute the weighted mean infiltration rate
!$ sum over cells from all watersheds in repository footprint
 TEMPSUM1 = NCELLSE1+NCELLSE2+NCELLSE3+NCELLSE4+NCELLSE5
 TEMPSUM2 = NCELLSE6+NCELLSE7+NCELLSE8+NCELLSE9+NCELSE10
 TEMPSUM3 = NCELLWT2+NCELLDH4+NCELLDH3+NCELLCW1+NCELLSW1
 SUMCELLS= MAKEGLOB(TEMPSUM1+TEMPSUM2+TEMPSUM3+NCELLWW1+NCELLUS1)
!$
 TMPMEAN1 =  NCELLWT2*INF_WT2 + NCELLDH4*INF_DH4 + NCELLDH3*INF_DH3
 TMPMEAN2 =  NCELLCW1*INF_CW1 + NCELLSW1*INF_SW1 + NCELLWW1*INF_WW1
 TMPMEAN3 =  NCELLSE1*INF_SE1 + NCELLSE2*INF_SE2 + NCELLSE3*INF_SE3
 TMPMEAN4 =  NCELLSE4*INF_SE4 + NCELLSE5*INF_SE5 + NCELLSE6*INF_SE6
 TMPMEAN5 =  NCELLSE7*INF_SE7 + NCELLSE8*INF_SE8 + NCELLSE9*INF_SE9
 TMPMEAN6 =  NCELSE10*INF_SE10 + NCELLUS1*INF_US1
 TMPSUM1 = TMPMEAN1 + TMPMEAN2 + TMPMEAN3
 TMPSUM2 = TMPMEAN4 + TMPMEAN5 + TMPMEAN6
 INFMEAN = MAKEGLOB(TMPSUM1 + TMPSUM2)/SUMCELLS
 EXIT
```

ATTACHMENT VIII

SAMPLE PROCEDURE FILES

This attachment contains listings of sample output files as produced in this analysis for each of the routines listed below (in order of use in analysis).

MATSET
PRELHS
POSTLHS
PREINFIL
POSTINFIL
ALGEBRACDB

Table VIII-1. LHS procedure file for Glacial Transition Climate.

```
$! -----
$! MODULE:  LHS.COM
$! PURPOSE: RUN YMP 1999 LATIN HYPERCUBE SAMPLING USING CMS
$!              VERSION OF LHS for Glacial Transition Climate
$! AUTHORS: Ron McCurley  COMPA/SNL  505-299-1282
$! DATE:    10/21/99
$! -----
$!
$! Setup for LHS run:
$! -----
$  SET NOON
$  INDIR := "I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.GLACIAL]"
$  OUTDIR := "I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.GLACIAL]"
$! Define file names
$  DEFINE LHS2_UIF$INPUT   'INDIR'LHS_1.INP
$  DEFINE LHS2_OUT$OUTPUT  'OUTDIR'LHS_INFIL1.OUT
$  DEFINE LHS2_DBG$OUTPUT  'OUTDIR'LHS1.DBG
$  DEFINE LHS2_NO2$SCRTCH  'OUTDIR'FOR002.DAT
$  DEFINE LHS2_NO3$SCRTCH  'OUTDIR'FOR003.DAT
$  DEFINE LHS2_NO4$SCRTCH  'OUTDIR'FOR004.DAT
$  DEFINE LHS2_NO7$SCRTCH  'OUTDIR'FOR007.DAT
$  DEFINE LHS2_NO8$SCRTCH  'OUTDIR'FOR008.DAT
$  DEFINE LHS2_NO9$SCRTCH  'OUTDIR'FOR009.DAT
$! -----
$!
$! Execute LHS:
$! -----
$  RUN WP$PRODROOT:[LHS.EXE]LHS_QA0250.EXE
$! -----
$!
$  DEFINE LHS2_UIF$INPUT   'INDIR'LHS_2.INP
$  DEFINE LHS2_OUT$OUTPUT  'OUTDIR'LHS_INFIL2.OUT
$  DEFINE LHS2_DBG$OUTPUT  'OUTDIR'LHS2.DBG
$  DEFINE LHS2_NO2$SCRTCH  'OUTDIR'FOR002.DAT
$  DEFINE LHS2_NO3$SCRTCH  'OUTDIR'FOR003.DAT
$  DEFINE LHS2_NO4$SCRTCH  'OUTDIR'FOR004.DAT
$  DEFINE LHS2_NO7$SCRTCH  'OUTDIR'FOR007.DAT
$  DEFINE LHS2_NO8$SCRTCH  'OUTDIR'FOR008.DAT
$  DEFINE LHS2_NO9$SCRTCH  'OUTDIR'FOR009.DAT
$! -----
$!
$! Execute LHS:
$! -----
$  RUN WP$PRODROOT:[LHS.EXE]LHS_QA0250.EXE
$! -----
$  DEFINE LHS2_UIF$INPUT   'INDIR'LHS_3.INP
$  DEFINE LHS2_OUT$OUTPUT  'OUTDIR'LHS_INFIL3.OUT
$  DEFINE LHS2_DBG$OUTPUT  'OUTDIR'LHS3.DBG
$  DEFINE LHS2_NO2$SCRTCH  'OUTDIR'FOR002.DAT
$  DEFINE LHS2_NO3$SCRTCH  'OUTDIR'FOR003.DAT
$  DEFINE LHS2_NO4$SCRTCH  'OUTDIR'FOR004.DAT
$  DEFINE LHS2_NO7$SCRTCH  'OUTDIR'FOR007.DAT
$  DEFINE LHS2_NO8$SCRTCH  'OUTDIR'FOR008.DAT
```

```
$  DEFINE LHS2_NO9$SCRTCH  'OUTDIR'FOR009.DAT
$! -----
$!
$! Execute LHS:
$! -----
$  RUN WP$PRODROOT:[LHS.EXE]LHS_QA0250.EXE
$! -----
$!
$! Inform user where output files are
$  write sys$output -
" Run completed"
$  write sys$output -
" The LHS output file is in ''OUTDIR'LHS_TEST.TRN"
$  write sys$output -
" The LHS tabular output file is in ''OUTDIR'LHS.DBG"
$! -----
$!
$ EXIT
```

Table VIII-2. Procedure file for MATSET 9.01.

```
!=================================================================
!
!   TITLE:    MATSET Procedure file for MATSET 9.01
!   ANALYST:  Ron McCurley - SNL/COMPA
!   DATE:     May-1999
!   Use Description: MATSET [Input CDB File (from GENMESH)] -
!                           [Input User Control File] -
!                           [Input SDB File (not used)] -
!                           [Calculation Type of SDB(not used)] -
!                           [Output CDB file] [Output Debug File]
!SDB is acronym for Secondary Database
!=================================================================
$MATSET GENMESH.CDB -
        MS_NOSDB.INP -
        CANCEL -
        CANCEL -
        LHS_NOSDB.CDB -
        MS_NOSDB.DBG
```

Table VIII-3. Procedure file for POSTLHS. Ron McCurley (299-1282) December 1999.

```
!=================================================================
!
!   TITLE:     POSTLHS Procedure file for POSTLHS 9.01
!   ANALYST:   Ron McCurley - SNL/COMPA
!   DATE:      May-1999
!   Use Description: POSTLHS [Input CDB File (from MATSET)] -
!                            [Input User Control File] -
!                            [Input LHS (output) Transfer File] -
!                            [Output CDB file] [Output Debug File]
!=================================================================
SET DEF I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.GLACIAL]
!
$ POSTLHS  I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.MATSET]LHS_NOSDB.CDB -
           POSTLHS.INP LHS_INFIL1.OUT POSTLHS_1_r###.CDB POSTLHS.DBG
$ POSTLHS  I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.MATSET]LHS_NOSDB.CDB -
           POSTLHS.INP LHS_INFIL2.OUT POSTLHS_2_r###.CDB POSTLHS.DBG
$ POSTLHS  I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.MATSET]LHS_NOSDB.CDB -
           POSTLHS.INP LHS_INFIL3.OUT POSTLHS_3_r###.CDB POSTLHS.DBG
```

Table VIII-4. PREINFIL procedure file – DEC alpha version

```
! Analyst--Ron McCurley SNL/COMPA  299-1282
! Date Oct 21, 1999.
! NEWPREINF [Input User Control File] [Input CDB File] -
!           [Input Template INFIL Input Control File] -
!           [Output INFIL Input Control File] [Output Debug File]
!
$ SET DEFAULT I1:[RDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.PrEINFIL.glacial]
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r001.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r001.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r002.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r002.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r003.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r003.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r004.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r004.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r005.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r005.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r006.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r006.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r007.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r007.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r008.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r008.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r009.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r009.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r010.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r010.INP -
        preinf.dbg
```

```
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r011.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r011.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r012.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r012.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r013.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r013.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r014.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r014.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r015.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r015.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r016.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r016.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r017.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r017.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r018.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r018.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r019.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r019.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r020.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r020.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r021.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r021.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r022.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r022.inp -
```

```
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r023.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r023.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r024.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r024.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r025.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r025.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r026.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r026.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r027.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r027.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r028.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r028.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r029.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r029.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r030.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r030.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r031.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r031.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r032.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r032.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r033.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r033.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r034.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
```

```
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r034.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r035.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r035.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r036.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r036.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r037.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r037.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r038.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r038.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r039.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r039.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r040.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r040.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r041.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r041.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r042.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r042.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r043.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r043.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r044.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r044.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r045.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r045.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r046.CDB -
```

```
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r046.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r047.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r047.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r048.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r048.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r049.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r049.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r050.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r050.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r051.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r051.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r052.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r052.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r053.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r053.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r054.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r054.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r055.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r055.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r056.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r056.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r057.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r057.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
```

```
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r058.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r058.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r059.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r059.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r060.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r060.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r061.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r061.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r062.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r062.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r063.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r063.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r064.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r064.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r065.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r065.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r066.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r066.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r067.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r067.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r068.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r068.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r069.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2a1_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r069.INP -
        preinf.dbg
$ NEWPrEINF  -
```

```
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r070.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r070.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r071.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r071.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r072.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r072.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r073.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r073.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r074.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r074.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r075.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r075.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r076.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r076.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r077.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r077.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r078.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r078.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r079.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r079.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r080.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r080.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r081.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r081.inp -
        preinf.dbg
```

```
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r082.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r082.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r083.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r083.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r084.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r084.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r085.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r085.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r086.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r086.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r087.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r087.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r088.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r088.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r089.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r089.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r090.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r090.INP -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r091.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r091.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r092.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r092.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r093.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r093.inp -
```

```
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r094.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r094.inp -
        preinf.dbg
$ newpreinf  -
        preinfil_ww1.inp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.lhs.glacial]postlhs_1_r095.cdb -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial]infil2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]infil_ww1_r095.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r096.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r096.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r097.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r097.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r098.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r098.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r099.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r099.INP -
        preinf.dbg
$ NEWPrEINF  -
        PrEINFIL_ww1.INP -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.lhs.glacial]POSTlhs_1_r100.CDB -
        I1:[rDMCCUr.INEEL_PA98.YMP_99.INFILTrATION.infil.glacial]INFIL2al_ww1.tmp -
        i1:[rdmccur.ineel_pa98.ymp_99.infiltration.infil.glacial.ww1]INFIL_ww1_r100.INP -
        preinf.dbg
```

Table VIII-5. Procedure file for POSTINFIL; 100 realizations. Ron McCurley (299-1282) December 1999.

```
!=====================================================================
! Procedure file to execute POSTINFIL over 100 realizations
!  ANALYST:  Ron McCurley, Oct 22, 1999
!  NEWPOSTINF [Input CDB file] [Input User Control File] -
!             [Input INFIL (Output) Transfer Main File] -
!             [Input INFIL (Output) Transfer Summary File] -
!             [Output CDB File] [Output Debug File]
!=====================================================================
!
$ SET DEFAULT I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.glacial]
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R001.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R001.OUT
-
    CANCEL -
    POSTINF_ww1_R001.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R002.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R002.OUT
-
    CANCEL -
    POSTINF_ww1_R002.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R003.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R003.OUT
-
    CANCEL -
    POSTINF_ww1_R003.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R004.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R004.OUT
-
    CANCEL -
    POSTINF_ww1_R004.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R005.CDB
-
```

```
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R005.OUT
-
    CANCEL -
    POSTINF_ww1_R005.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R006.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R006.OUT
-
    CANCEL -
    POSTINF_ww1_R006.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R007.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R007.OUT
-
    CANCEL -
    POSTINF_ww1_R007.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R008.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R008.OUT
-
    CANCEL -
    POSTINF_ww1_R008.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R009.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R009.OUT
-
    CANCEL -
    POSTINF_ww1_R009.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R010.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R010.OUT
-
    CANCEL -
    POSTINF_ww1_R010.CDB -
    POSTINF.DBG
```

```
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R011.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R011.OUT
-
    CANCEL -
    POSTINF_ww1_R011.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R012.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R012.OUT
-
    CANCEL -
    POSTINF_ww1_R012.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R013.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R013.OUT
-
    CANCEL -
    POSTINF_ww1_R013.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R014.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R014.OUT
-
    CANCEL -
    POSTINF_ww1_R014.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R015.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R015.OUT
-
    CANCEL -
    POSTINF_ww1_R015.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R016.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R016.OUT
-
```

```
    CANCEL -
    POSTINF_ww1_R016.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R017.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R017.OUT
-
    CANCEL -
    POSTINF_ww1_R017.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R018.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R018.OUT
-
    CANCEL -
    POSTINF_ww1_R018.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R019.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R019.OUT
-
    CANCEL -
    POSTINF_ww1_R019.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R020.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R020.OUT
-
    CANCEL -
    POSTINF_ww1_R020.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R021.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R021.OUT
-
    CANCEL -
    POSTINF_ww1_R021.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R022.CDB
-
    POSTINF_ww1.INP -
```

```
I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R022.OUT
-
    CANCEL -
    POSTINF_ww1_R022.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R023.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R023.OUT
-
    CANCEL -
    POSTINF_ww1_R023.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R024.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R024.OUT
-
    CANCEL -
    POSTINF_ww1_R024.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R025.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R025.OUT
-
    CANCEL -
    POSTINF_ww1_R025.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R026.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R026.OUT
-
    CANCEL -
    POSTINF_ww1_R026.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R027.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R027.OUT
-
    CANCEL -
    POSTINF_ww1_R027.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
```

```
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R028.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R028.OUT
-
    CANCEL -
    POSTINF_ww1_R028.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R029.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R028.OUT
-
    CANCEL -
    POSTINF_ww1_R029.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R030.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R030.OUT
-
    CANCEL -
    POSTINF_ww1_R030.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R031.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R031.OUT
-
    CANCEL -
    POSTINF_ww1_R031.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R032.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R032.OUT
-
    CANCEL -
    POSTINF_ww1_R032.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R033.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R033.OUT
-
    CANCEL -
```

```
    POSTINF_ww1_R033.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R034.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R034.OUT
-
    CANCEL -
    POSTINF_ww1_R034.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R035.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R035.OUT
-
    CANCEL -
    POSTINF_ww1_R035.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R036.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R036.OUT
-
    CANCEL -
    POSTINF_ww1_R036.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R037.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R037.OUT
-
    CANCEL -
    POSTINF_ww1_R037.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R038.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R038.OUT
-
    CANCEL -
    POSTINF_ww1_R038.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R039.CDB
-
    POSTINF_ww1.INP -
```

```
I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R039.OUT
-
    CANCEL -
    POSTINF_ww1_R039.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R040.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R040.OUT
-
    CANCEL -
    POSTINF_ww1_R040.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R041.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R041.OUT
-
    CANCEL -
    POSTINF_ww1_R041.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R042.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R042.OUT
-
    CANCEL -
    POSTINF_ww1_R042.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R043.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R043.OUT
-
    CANCEL -
    POSTINF_ww1_R043.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R044.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R044.OUT
-
    CANCEL -
    POSTINF_ww1_R044.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
```

```
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R045.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R045.OUT
-
    CANCEL -
    POSTINF_ww1_R045.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R046.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R046.OUT
-
    CANCEL -
    POSTINF_ww1_R046.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R047.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R047.OUT
-
    CANCEL -
    POSTINF_ww1_R047.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R048.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R048.OUT
-
    CANCEL -
    POSTINF_ww1_R048.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R049.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R049.OUT
-
    CANCEL -
    POSTINF_ww1_R049.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R050.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R050.OUT
-
    CANCEL -
```

```
    POSTINF_ww1_R050.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R051.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R051.OUT
-
    CANCEL -
    POSTINF_ww1_R051.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R052.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R052.OUT
-
    CANCEL -
    POSTINF_ww1_R052.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R053.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R053.OUT
-
    CANCEL -
    POSTINF_ww1_R053.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R054.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R054.OUT
-
    CANCEL -
    POSTINF_ww1_R054.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R055.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R055.OUT
-
    CANCEL -
    POSTINF_ww1_R055.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R056.CDB
-
    POSTINF_ww1.INP -
```

```
I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R056.OUT
-
    CANCEL -
    POSTINF_ww1_R056.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R057.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R057.OUT
-
    CANCEL -
    POSTINF_ww1_R057.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R058.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R058.OUT
-
    CANCEL -
    POSTINF_ww1_R058.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R059.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R059.OUT
-
    CANCEL -
    POSTINF_ww1_R059.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R060.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R060.OUT
-
    CANCEL -
    POSTINF_ww1_R060.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R061.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R061.OUT
-
    CANCEL -
    POSTINF_ww1_R061.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
```

```
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R062.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R062.OUT
-
    CANCEL -
    POSTINF_ww1_R062.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R063.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R063.OUT
-
    CANCEL -
    POSTINF_ww1_R063.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R064.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R064.OUT
-
    CANCEL -
    POSTINF_ww1_R064.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R065.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R065.OUT
-
    CANCEL -
    POSTINF_ww1_R065.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R066.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R066.OUT
-
    CANCEL -
    POSTINF_ww1_R066.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R067.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R067.OUT
-
    CANCEL -
```

```
    POSTINF_ww1_R067.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R068.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R068.OUT
-
    CANCEL -
    POSTINF_ww1_R068.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R069.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R069.OUT
-
    CANCEL -
    POSTINF_ww1_R069.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R070.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R070.OUT
-
    CANCEL -
    POSTINF_ww1_R070.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R071.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R071.OUT
-
    CANCEL -
    POSTINF_ww1_R071.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R072.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R072.OUT
-
    CANCEL -
    POSTINF_ww1_R072.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R073.CDB
-
    POSTINF_ww1.INP -
```

```
I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R073.OUT
-
    CANCEL -
    POSTINF_ww1_R073.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R074.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R074.OUT
-
    CANCEL -
    POSTINF_ww1_R074.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R075.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R075.OUT
-
    CANCEL -
    POSTINF_ww1_R075.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R076.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R076.OUT
-
    CANCEL -
    POSTINF_ww1_R076.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R077.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R077.OUT
-
    CANCEL -
    POSTINF_ww1_R077.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R078.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R078.OUT
-
    CANCEL -
    POSTINF_ww1_R078.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
```

```
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R079.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R078.OUT
-
    CANCEL -
    POSTINF_ww1_R079.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R080.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R080.OUT
-
    CANCEL -
    POSTINF_ww1_R080.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R081.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R081.OUT
-
    CANCEL -
    POSTINF_ww1_R081.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R082.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R082.OUT
-
    CANCEL -
    POSTINF_ww1_R082.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R083.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R083.OUT
-
    CANCEL -
    POSTINF_ww1_R083.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R084.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R084.OUT
-
    CANCEL -
```

```
    POSTINF_ww1_R084.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R085.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R085.OUT
-
    CANCEL -
    POSTINF_ww1_R085.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R086.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R086.OUT
-
    CANCEL -
    POSTINF_ww1_R086.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R087.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R087.OUT
-
    CANCEL -
    POSTINF_ww1_R087.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R088.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R088.OUT
-
    CANCEL -
    POSTINF_ww1_R088.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R089.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R089.OUT
-
    CANCEL -
    POSTINF_ww1_R089.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R090.CDB
-
    POSTINF_ww1.INP -
```

```
I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R090.OUT
-
    CANCEL -
    POSTINF_ww1_R090.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R091.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R091.OUT
-
    CANCEL -
    POSTINF_ww1_R091.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R092.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R092.OUT
-
    CANCEL -
    POSTINF_ww1_R092.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R093.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R093.OUT
-
    CANCEL -
    POSTINF_ww1_R093.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R094.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R094.OUT
-
    CANCEL -
    POSTINF_ww1_R094.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R095.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R095.OUT
-
    CANCEL -
    POSTINF_ww1_R095.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
```

```
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R096.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R096.OUT
-
    CANCEL -
    POSTINF_ww1_R096.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R097.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R097.OUT
-
    CANCEL -
    POSTINF_ww1_R097.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R098.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R098.OUT
-
    CANCEL -
    POSTINF_ww1_R098.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R099.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R099.OUT
-
    CANCEL -
    POSTINF_ww1_R099.CDB -
    POSTINF.DBG
$ NEWPOSTINF -
    I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.glacial]POSTLHS_1_R100.CDB
-
    POSTINF_ww1.INP -

I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.INFIL.glacial.ww1]INFILSM_R100.OUT
-
    CANCEL -
    POSTINF_ww1_R100.CDB -
    POSTINF.DBG
```

Table VIII-6. Procedure file for PRELHS for Glacial Transition Climate

```
$! PURPOSE: RUN USING A "LOCAL BUILD"  EXECUTABLE for PRELHS
$! AUTHOR:  Ron McCurley
$! DATE:    10/14/99.
$! -----
$!
$  SET NOON
$  LHSDB   := INFIL.SDB
$!  LHSCALC :=
$  OUTDIR  := "I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.PRELHS.GLACIAL]"
$  TRNDIR  := "I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.GLACIAL]"
$  EXE_SRC :==
$I1:[RDMCCUR.INEEL_PA98.YMP_99.MONTE_CARLO.PRELHS.SOURCE_CODE]INELPRELHS.EXE
$! -----
$!
$! Define file names
$! -----
$  DEFINE LHS1_NO2$SCRATCH FOR002.DAT
$  DEFINE LHS2_NO3$SCRATCH FOR003.DAT
$! -----
$!
$! Execute test case COUNT
$! -----
$  EXE_SRC PRELHS_1.INP  -
           INFIL.SDB -
           'TRNDIR'LHS_1.INP   -
           'OUTDIR'PRELHS.DBG
$! -----
$  EXE_SRC PRELHS_2.INP  -
           INFIL.SDB -
           'TRNDIR'LHS_2.INP   -
           'OUTDIR'PRELHS.DBG
$! -----
$  EXE_SRC PRELHS_3.INP  -
           INFIL.SDB -
           'TRNDIR'LHS_3.INP   -
           'OUTDIR'PRELHS.DBG
$! -----
$!
$! Inform user where output files are
$  write sys$output -
" Test run completed"
$  write sys$output -
" The debug output file is in ''OUTDIR'PRELHS_TEST.DBG"
$  write sys$output -
" The LHS Input file is in ''TRNDIR'LHS_TEST.INP"
$! -----
$!
$ EXIT
```

Table VIII-7. Sample procedure file for ALGEBRACDB, 100 realizations. McCurley (299-1282) December 1999.

```
!=====================================================================
!Procedure file to execute ALGEBRACDB over 100 realizations
!  ANALYST:  Ron McCurley, Oct 22, 1999
! ALGEBRA [Input CDB] [Output CDB] [User Control File] [Debug File]
!=====================================================================
!
$ SET DEFAULT I1:[RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.GLACIAL]
$ ALGEBRA  POSTINF_US1_R001.CDB MEAN_R001.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R002.CDB MEAN_R002.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R003.CDB MEAN_R003.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R004.CDB MEAN_R004.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R005.CDB MEAN_R005.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R006.CDB MEAN_R006.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R007.CDB MEAN_R007.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R008.CDB MEAN_R008.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R009.CDB MEAN_R009.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R010.CDB MEAN_R010.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R011.CDB MEAN_R011.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R012.CDB MEAN_R012.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R013.CDB MEAN_R013.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R014.CDB MEAN_R014.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R015.CDB MEAN_R015.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R016.CDB MEAN_R016.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R017.CDB MEAN_R017.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R018.CDB MEAN_R018.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R019.CDB MEAN_R019.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R020.CDB MEAN_R020.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R021.CDB MEAN_R021.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R022.CDB MEAN_R022.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R023.CDB MEAN_R023.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R024.CDB MEAN_R024.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R025.CDB MEAN_R025.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R026.CDB MEAN_R026.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R027.CDB MEAN_R027.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R028.CDB MEAN_R028.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R029.CDB MEAN_R029.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R030.CDB MEAN_R030.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R031.CDB MEAN_R031.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R032.CDB MEAN_R032.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R033.CDB MEAN_R033.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R034.CDB MEAN_R034.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R035.CDB MEAN_R035.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R036.CDB MEAN_R036.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R037.CDB MEAN_R037.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R038.CDB MEAN_R038.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R039.CDB MEAN_R039.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R040.CDB MEAN_R040.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R041.CDB MEAN_R041.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R042.CDB MEAN_R042.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R043.CDB MEAN_R043.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R044.CDB MEAN_R044.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R045.CDB MEAN_R045.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R046.CDB MEAN_R046.CDB ALG_MEAN.INP ALG.DBG
```

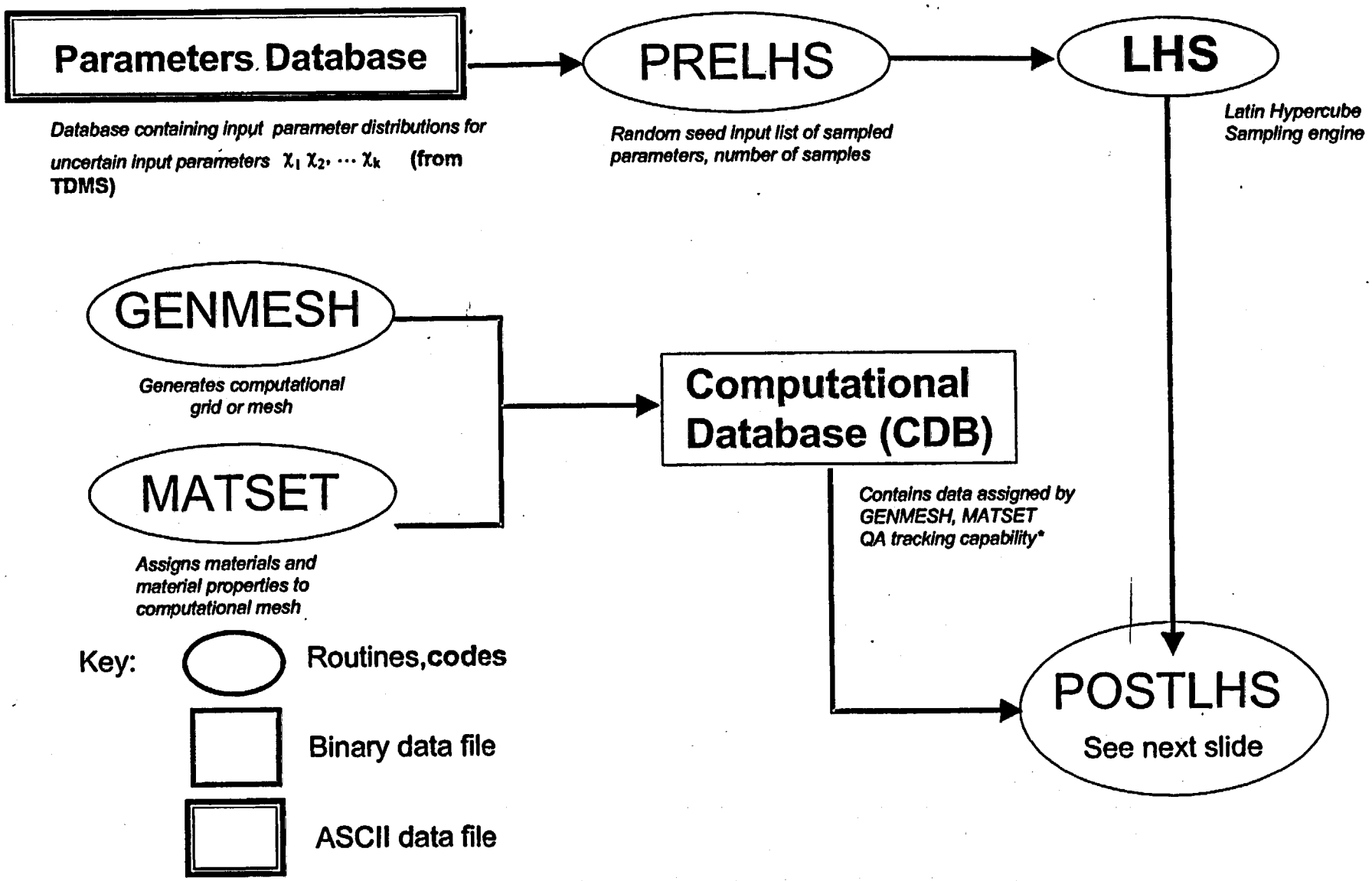
```
$ ALGEBRA  POSTINF_US1_R047.CDB MEAN_R047.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R048.CDB MEAN_R048.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R049.CDB MEAN_R049.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R050.CDB MEAN_R050.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R051.CDB MEAN_R051.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R052.CDB MEAN_R052.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R053.CDB MEAN_R053.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R054.CDB MEAN_R054.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R055.CDB MEAN_R055.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R056.CDB MEAN_R056.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R057.CDB MEAN_R057.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R058.CDB MEAN_R058.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R059.CDB MEAN_R059.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R060.CDB MEAN_R060.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R061.CDB MEAN_R061.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R062.CDB MEAN_R062.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R063.CDB MEAN_R063.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R064.CDB MEAN_R064.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R065.CDB MEAN_R065.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R066.CDB MEAN_R066.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R067.CDB MEAN_R067.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R068.CDB MEAN_R068.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R069.CDB MEAN_R069.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R070.CDB MEAN_R070.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R071.CDB MEAN_R071.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R072.CDB MEAN_R072.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R073.CDB MEAN_R073.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R074.CDB MEAN_R074.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R075.CDB MEAN_R075.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R076.CDB MEAN_R076.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R077.CDB MEAN_R077.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R078.CDB MEAN_R078.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R079.CDB MEAN_R079.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R080.CDB MEAN_R080.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R081.CDB MEAN_R081.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R082.CDB MEAN_R082.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R083.CDB MEAN_R083.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R084.CDB MEAN_R084.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R085.CDB MEAN_R085.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R086.CDB MEAN_R086.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R087.CDB MEAN_R087.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R088.CDB MEAN_R088.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R089.CDB MEAN_R089.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R090.CDB MEAN_R090.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R091.CDB MEAN_R091.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R092.CDB MEAN_R092.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R093.CDB MEAN_R093.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R094.CDB MEAN_R094.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R095.CDB MEAN_R095.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R096.CDB MEAN_R096.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R097.CDB MEAN_R097.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R098.CDB MEAN_R098.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R099.CDB MEAN_R099.CDB ALG_MEAN.INP ALG.DBG
$ ALGEBRA  POSTINF_US1_R100.CDB MEAN_R100.CDB ALG_MEAN.INP ALG.DBG
```

ATTACHMENT IX

FLOW DIAGRAM FOR UNCERTAINTY ANALYSIS

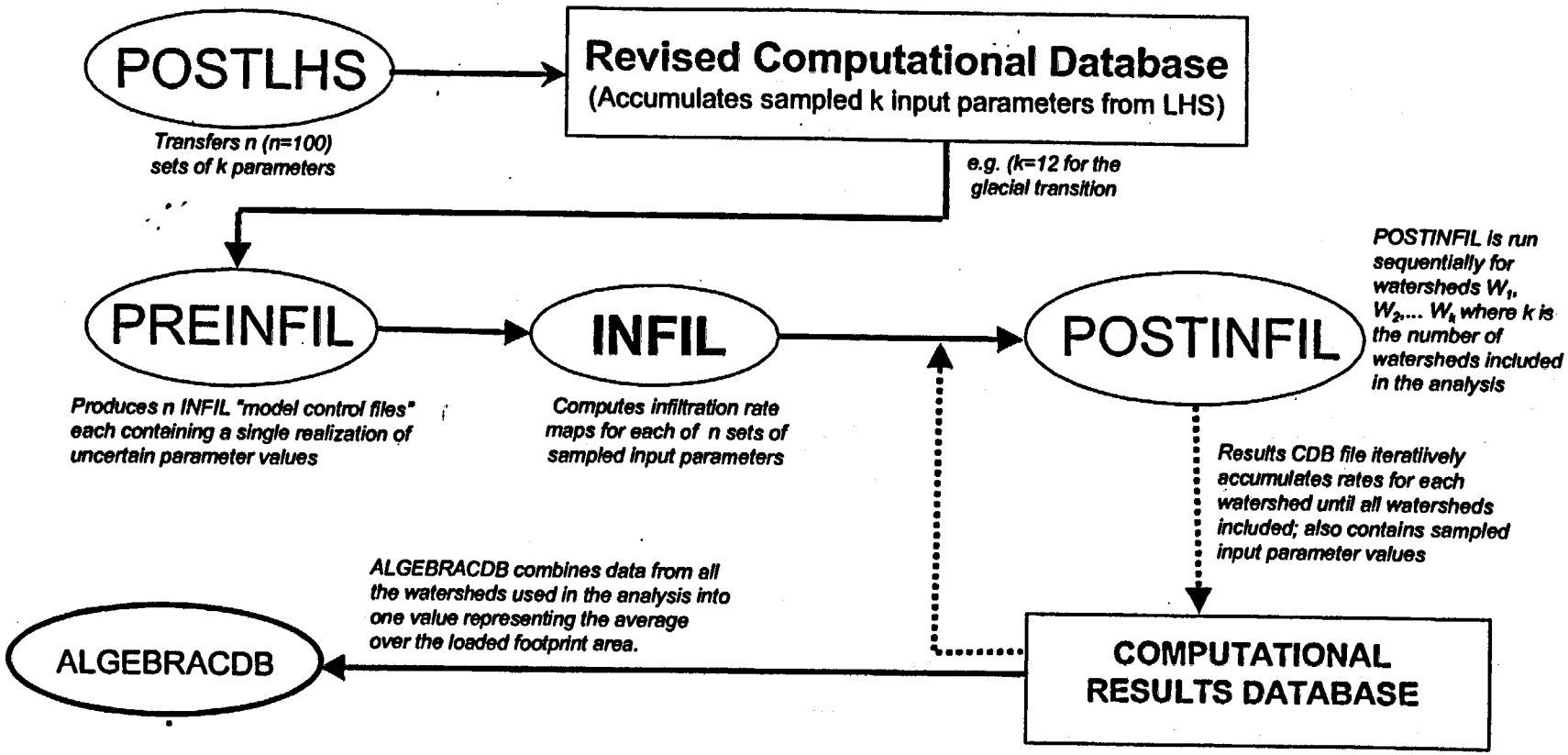
This attachment contains a diagram showing the flow of information as the uncertainty analysis proceeds towards the final product of a set of 100 realizations resulting in an uncertainty distribution for the infiltration rate maps as spatially averaged over the modeled rectangular region.

Analysis Flow Diagram



* applies to all CDB files generated

Continue to next slide


POSTLHS
Transfers n (n=100) sets of k parameters
Revised Computational Database
(Accumulates sampled k input parameters from LHS)
e.g. (k=12 for the glacial transition
PREINFIL
Produces n INFIL "model control files" each containing a single realization of uncertain parameter values
INFIL
Computes infiltration rate maps for each of n sets of sampled input parameters
POSTINFIL
POSTINFIL is run sequentially for watersheds W_1, W_2,... W_k where k is the number of watersheds included in the analysis
Results CDB file iteratively accumulates rates for each watershed until all watersheds included; also contains sampled input parameter values
COMPUTATIONAL RESULTS DATABASE
ALGEBRACDB combines data from all the watersheds used in the analysis into one value representing the average over the loaded footprint area.
ALGEBRACDB

ATTACHMENT X

DOCUMENTATION FOR OTHER ROUTINES USED IN THE UNCERTAINTY ANALYSIS

(TO SATISFY AP-SI.1Q)

Documentation of software routines to satisfy AP-SI.1Q, Paragraph 5.1.1.2. These are all routines because they can be verified by visual inspection or hand calculations. The routines documented here are

ALGEBRACDB
GENMESH
MATSET
PRELHS
POSTLHS

ALGEBRACDB Version 2.35 Software Routine Verification

MODIFICATIONS OF FE/FD COMPUTATIONAL DATABASE

involving the following:
(1) NEW VARIABLES ASSIGNED VALUES ACCORDING TO ALGEBRAIC EQUATION OR FUNCTION

Documentation to satisfy first and second bullets of Section 5.1.1.2:

A software routine was used in accordance with AP-SI.1Q, Section 5.1.1.2 for the purpose of algebraically manipulating appropriate parameters and variables contained in a Computational Database file (CDB) to create appropriate new variables. These new variables and the corresponding algebraic equation are defined by the analyst in an input user control file (see Attachment I-25). The software developed, ALGEBRACDB, version 2.35, is in FORTRAN 77. The executable and the change history reside in the following directory locations on a DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories:

WP$PRODROOT:[ALG.EXE]ALGEBRACDB_PA96.EXE

DISK$BEATLE_CCA1:[NON_HSM.CMS_WIPP_PA.ALG]01CMS.HIS;1

This satisfies the requirements first and second bullets of Section 5.1.1.2 of procedure AP-SI.1Q for software routines.

Documentation to satisfy third bullet of Section 5.1.1.2:

ALGEBRACDB capability requirements, including acceptance criteria, and tests for verification and validation and can be found in the Requirements Document, Verification and Validation Plan (SNL, 1995b) and the Validation Document (SNL, 1995c). The use of the routine in this AMR was confined to algebraic manipulations involving addition, multiplication, and calculation of an arithmetic mean. The use of this routine in this manner falls within the domain of the use tested in the documentation referred to above. For this analysis, hand calculations were done verifying that the calculations done by ALGEBRACDB (see Table X-1) to obtain the spatially averaged infiltration rate for the modeled rectangular region (Figure 1-1) are correct. The sum of the cells over all the watersheds is shown as **SUMCELLS**. In italics are temporary partial sums of terms which are a product of average infiltration rate and number of cells for each watershed (see Table VII-11, the ALGEBRACDB input file used to produce this file displayed in Table X-1). In bold Italics, ***INFMEAN*** represents the averaged infiltration rate over the modeled rectangular region.

Table X-1. GROPECDB Listing of Realization 001 Showing Values of Number of Cells (bolded) Within Modeled Rectangular Region and Corresponding Average Infiltration Rates (adjacent to right).

```
                                GROPECDB_PA96

 GGGGG  RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB         PPPPPP
GG  GGG RRRRR   OO   OO PP      EE      CC      DD   DD BB   BB         PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB         PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB _______ PP

                      GROPECDB_PA96 Version 2.12
                       PROD PA96 Built 06/27/96
                       Sponsored by Amy Gilkey

                      Run on 04/05/00 at 09:14:19
                   Run on ALPHA AXP BEATLE OpenVMS V7.2-1

Database:  I1:[000000.RDMCCUR.YMP99.INFILTRATION.POSTINFIL.GLACIAL.NEW]MEAN_R001.CDB;1
Written on:  12/09/99 12:14:46

CAMDAT Version:   1        (EXODUS Version:   1)

GLOBAL TIME STEP VARIABLES
```

Step	Time				
		NCELLSE1	INF_SE1	RUN_SE1	**NCELLSE2**
		INF_SE2	RUN_SE2	**NCELLSE3**	INF_SE3
		RUN_SE3	**NCELLSE4**	INF_SE4	RUN_SE4
		NCELLSE5	INF_SE5	RUN_SE5	**NCELLSE6**
		INF_SE6	RUN_SE6	**NCELLSE7**	INF_SE7
		RUN_SE7	**NCELLSE8**	INF_SE8	RUN_SE8
		NCELLSE9	INF_SE9	RUN_SE9	**NCELSE10**
		INF_SE10	RUN_SE10	**NCELLWT2**	INF_WT2
		RUN_WT2	**NCELLDH4**	INF_DH4	RUN_DH4
		NCELLDH3	INF_DH3	RUN_DH3	**NCELLCW1**
		INF_CW1	RUN_CW1	**NCELLSW1**	INF_SW1
		RUN_SW1	**NCELLWW1**	INF_WW1	RUN_WW1
		NCELLUS1	INF_US1	RUN_US1	TEMPSUM1
		TEMPSUM2	TEMPSUM3	**SUMCELLS**	*TMPMEAN1*
		TMPMEAN2	*TMPMEAN3*	*TMPMEAN4*	*TMPMEAN5*
		TMPMEAN6	*TMPSUM1*	*TMPSUM2*	***INFMEAN***
1	0.00000E+00	**6.10000E+01**	7.59469E+01	0.00000E+00	**5.00000E+00**
		4.90990E+01	0.00000E+00	**4.70000E+01**	1.23239E+02
		0.00000E+00	**8.20000E+01**	1.01059E+02	0.00000E+00
		4.10000E+01	7.21075E+01	0.00000E+00	**1.14000E+02**
		9.24462E+01	0.00000E+00	**6.70000E+01**	7.74080E+01
		0.00000E+00	**7.00000E+01**	8.63004E+01	0.00000E+00
		8.50000E+01	8.58878E+01	0.00000E+00	**8.00000E+00**
		6.48678E+01	0.00000E+00	**1.58300E+03**	4.37066E+01
		0.00000E+00	**6.10000E+02**	4.03951E+01	0.00000E+00
		1.18800E+03	4.21285E+01	0.00000E+00	**4.17000E+02**
		5.63918E+01	0.00000E+00	**7.18000E+02**	4.74610E+01
		0.00000E+00	**8.75000E+02**	6.31565E+01	0.00000E+00
		6.86000E+02	4.92632E+01	0.00000E+00	2.36000E+02
		3.44000E+02	4.51600E+03	**6.65700E+03**	*1.43877E+05*
		1.12854E+05	*1.06705E+04*	*2.17821E+04*	*1.85278E+04*
		3.43135E+04	*2.67402E+05*	*7.46234E+04*	***5.13783E+01***

GENMESH Version 6.08 Software Routine Verification

CREATION OF FE/FD COMPUTATIONAL DATABASE

containing the following:
(2) ELEMENT BLOCKS (OR MATERIALS)

(3) DESIGNATED MATERIAL PROPERTY NAMES.

Documentation to satisfy first and second bullet of Section 5.1.1.2:

A software routine was used in accordance with AP-SI.1Q, Section 5.1.1.2 for the purpose of developing and assigning geometrical attributes to a computational mesh for input to MATSET to create a Computational Database file (CDB); with appropriate material properties as selected by the analyst. The software developed, GENMESH, version 6.08, is in FORTRAN 77. The executable and the change history reside in the following directory locations on a DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories:

WP$CGMROOT:[GM.CGM$001]GENMESH_QA0608.EXE;2

WP$CGMROOT:[GM]01CGM.HIS;1

This satisfies the requirements first and second bullets of Section 5.1.1.2 of procedure AP-SI.1Q for software routines.

Documentation to satisfy third bullet of Section 5.1.1.2:

GENMESH capability requirements, including acceptance criteria, and tests for verification and validation and can be found in the Requirements Document, Verification and Validation Plan (RDVVP) (SNL, 1996c) and the Validation Document (SNL, 1995d). GENMESH was used to create a binary file referred to in this AMR as a CDB file to serve as a vehicle for the traceable storage of data and computation steps as developed in the uncertainty analysis documented in this AMR. The CDB file is updated by MATSET (Table X-1) to prepare for the sampling process. The use of this routine in this manner falls within the domain of the use tested in the documentation referred to above. A visual inspection of the GROPECDB display of this file (Table X-1) verifies the use of this routine (in conjunction with MATSET).

MATSET Version 9.01 Software Routine Verification

CREATION OF FE/FD COMPUTATIONAL DATABASE

containing the following:
(4) ELEMENT BLOCKS (OR MATERIALS)

(5) DESIGNATED MATERIAL PROPERTY NAMES.

Documentation to satisfy first and second bullet of Section 5.1.1.2:

A software routine was used in accordance with AP-SI.1Q, Section 5.1.1.2 for the purpose of assigning material properties and corresponding (default) values to a Computational Database file (CDB); these material properties are the selected candidate properties for sampling (using Latin Hypercube Sampling) The software developed, MATSET, Version 9.01, is in FORTRAN 77. The executable and the change history reside in the following directory locations on a DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories:

WP$CMSROOT:[MS.CMS$001]MATSET_QA0901.EXE;2

WP$CMSROOT:[MS]01CMS.HIS;1

This satisfies the requirements first and second bullets of Section 5.1.1.2 of procedure AP-SI.1Q for software routines.

Documentation to satisfy third bullet of Section 5.1.1.2:

MATSET capability requirements, including acceptance criteria, and tests for verification and validation and can be found in the Requirements Document, Verification and Validation Plan (SNL, 1996d) and the Validation Document (SNL, 1996e). MATSET was used for adding, to the CDB file created by GENMESH, material property names (bold in Table X-2) corresponding to the parameters being sampled as developed in the uncertainty analysis documented in this AMR. The use of this routine in this manner falls within the domain of the use tested in the documentation referred to above. A visual inspection of the GROPECDB display of this file verifies the use of this routine (in conjunction with GENMESH).

Table X-2. GROPECDB listing of binary CDB file containing properties to be sampled by LHS

```
                                GROPECDB_PA96

 GGGGG  RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB          PPPPPP
GG  GGG RRRRR   OO   OO PP      EE      CC      DD   DD BB   BB          PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB          PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB           PP

                        GROPECDB_PA96 Version 2.12
                         PROD PA96 Built 06/27/96
                         Sponsored by Amy Gilkey

                       Run on 04/05/00 at 09:02:13
                    Run on ALPHA AXP BEATLE OpenVMS V7.2-1

Database:  I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.MATSET]LHS_NOSDB.CDB;5
Written on:  11/09/99 18:22:47

CAMDAT Version:   1        (EXODUS Version:   1)

PROPERTIES

Element Block 1)    "FOOTPRNT"     1=ID    75 elements (1..75)
      8-node     5 attributes    13 properties
   BRPOROS      BRZDEPTH     SOILDEPM     PRECIPM      POTETMUL     BRPERM
   SOILPERM     ETCOEFFA     ETCOEFFB     FLAREA       ATEMP1       SNOPAR1
   SUBPAR1
  2.00000E-02  1.50000E+00  1.00000E+00  1.00000E+00  1.00000E+00  1.00000E+00
  1.00000E+00 -1.50000E+00  1.26000E+00  5.00000E-01  1.73000E+01  1.78000E+00
  1.00000E-01

Element Block 2)    "OUTSIDE "     2=ID    25 elements (76..100)
      8-node     5 attributes    13 properties
   BRPOROS      BRZDEPTH     SOILDEPM     PRECIPM      POTETMUL     BRPERM
   SOILPERM     ETCOEFFA     ETCOEFFB     FLAREA       ATEMP1       SNOPAR1
   SUBPAR1
  2.00000E-02  1.50000E+00  1.00000E+00  1.00000E+00  1.00000E+00  1.00000E+00
  1.00000E+00 -1.50000E+00  1.26000E+00  5.00000E-01  1.73000E+01  1.78000E+00
  1.00000E-01
```

PRELHS Version 2.02 Software Routine Verification

CREATION OF LHS INPUT FILE CONTAINING THE FOLLOWING:

DESIGNATED MATERIAL PROPERTY NAMES WITH CORRESPONDING DISTRIBUTION TYPE AND RANGE.

Documentation to satisfy first and second bullet of Section 5.1.1.2:

A software routine was used in accordance with AP-SI.1Q, Section 5.1.1.2 for the purpose of developing input files for LHS. The developed LHS input files contain distributions of 'material property' or 'attribute' parameters as selected by the analyst in a PRELHS user control (see Attachment I-6) and as defined in an input data file (see Attachment I-7). The software developed, PRELHS, Version 2.02, is in FORTRAN 77. The executable and the change history reside in the following directory locations on a DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories:

I1:[RDMCCUR.INEEL_PA98.YMP_99.MONTE_CARLO.PRELHS.SOURCE_CODE]INELPRELH.EXE

WP$CGMROOT:[LHS]01CGM.HIS;1

This satisfies the requirements first and second bullets of Section 5.1.1.2 of procedure AP-SI.1Q for software routines.

Documentation to satisfy third bullet of Section 5.1.1.2:

PRELHS capability requirements, including acceptance criteria, and tests for verification and validation and can be found in the Requirements Document, Verification and Validation Plan (SNL, 1997a) and the Validation Document (SNL, 1997b). PRELHS was used to create input files for LHS V. 2.50 with the defined distribution and range of the parameters chosen for sampling in this analysis. Visual verification can be made by comparing the distribution and ranges of parameters as found in Table VII-4b with the LHS input file found in Table VII-5. The use of this routine in this manner also falls within the domain of the use tested in the documentation referred to above.

POSTLHS Version 4.07 Software Routine Verification

MODIFICATION OF FE/FD COMPUTATIONAL DATABASE

containing the following:
(6) DESIGNATED MATERIAL PROPERTY NAMES.

(2) the assigned value for each sampled material property, by LHS, for each realization

Documentation to satisfy first and second bullets of Section 5.1.1.2:

A software routine was used in accordance with AP-SI.1Q, Section 5.1.1.2 for the purpose of and assigning material property (or attribute) values, from each realization, to a corresponding Computational Database file (CDB) for all realizations. These material property (or attribute) values are for appropriate material properties (or attributes) as selected by the analyst in a user control file (see Attachment I-8) and are assigned values as sampled by LHS. The software developed, POSTLHS, Version 4.07, is in FORTRAN 77. The executable and the change history reside in the following directory locations on a DEC ALPHA using the operating system open VMS AXP V7 at Sandia National Laboratories:

WP$CMSROOT:[LHS.CMS$001]POSTLHS_PA96.EXE;2

WP$CGMROOT:[LHS]01CGM.HIS;1

This satisfies the requirements first and second bullets of Section 5.1.1.2 of procedure AP-SI.1Q for software routines.

Documentation to satisfy third bullet of Section 5.1.1.2:

POSTLHS capability requirements, including acceptance criteria, and tests for verification and validation and can be found in the Requirements Document, Verification and Validation Plan (SNL, 1995e) and the Validation Document (SNL, 1995f). POSTLHS was used to add the sampled input (to INFIL V A_2.a1) parameter values (as found in LHS output) to a CDB file corresponding to each realization. The name of each CDB file included a designator indicating the realization that produced these values. The use of this routine in this manner falls within the domain of the use tested in the documentation referred to above. Visual comparison of the GROPECDB display of each CDB file (e.g. Table X-3 for realization 001) produced by POSTLHS with the values produced by LHS (Attachment XI) verifies that the appropriate values corresponding to each realization are correctly inserted into that CDB file.

Table X-3. GROPECDB listing of realization 001 showing values of sampled properties

```
GGGGG   RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB         PPPPPP
GG  GGG RRRRR   OO   OO PP      EE      CC      DD   DD BB   BB         PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB         PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB _______ PP
```

GROPECDB_PA96 Version 2.12
PROD PA96 Built 06/27/96
Sponsored by Amy Gilkey

Run on 04/05/00 at 09:08:45
Run on ALPHA AXP BEATLE OpenVMS V7.2-1

Database:
I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.LHS.GLACIAL]POSTLHS_1_R001.CDB;2
Written on: 12/01/99 18:02:24

CAMDAT Version: 1 (EXODUS Version: 1)

PROPERTIES

Element Block 1) "FOOTPRNT" 1=ID 75 elements (1..75)
8-node 5 attributes 13 properties

BRPOROS	BRZDEPTH	SOILDEPM	PRECIPM	POTETMUL	BRPERM
SOILPERM	ETCOEFFA	ETCOEFFB	FLAREA	ATEMP1	SNOPAR1
SUBPAR1					
2.95100E-02	3.74700E+00	1.12800E+00	1.11500E+00	9.01500E-01	5.48200E+00
2.22800E+00	1.06100E+01	1.10200E+00	3.04400E-01	1.73000E+01	2.15800E+00
1.57100E-01					

Element Block 2) "OUTSIDE " 2=ID 25 elements (76..100)
8-node 5 attributes 13 properties

BRPOROS	BRZDEPTH	SOILDEPM	PRECIPM	POTETMUL	BRPERM
SOILPERM	ETCOEFFA	ETCOEFFB	FLAREA	ATEMP1	SNOPAR1
SUBPAR1					
2.00000E-02	1.50000E+00	1.00000E+00	1.00000E+00	1.00000E+00	1.00000E+00
1.00000E+00	-1.50000E+00	1.26000E+00	5.00000E-01	1.73000E+01	1.78000E+00
1.00000E-01					

ATTACHMENT XI

OUTPUT FILE FROM LHS CONTAINING SAMPLED VALUES OF INPUT PARAMETERS FOR ALL 100 REALIZATIONS

This attachment contains sampled values for the twelve uncertain input parameters listed in Table 4-1 (as obtained from LHS) using 100 realizations.

Table XI-1. LHS Output File Containing Sampled values for 12 Uncertain Parameters

```
TITLE SDB Name = INFIL.SDB        , Ver = 1                12/01/99 17:58:05
RANDOM SEED =    733050198
NUMBER OF VARIABLES =  12
NUMBER OF OBSERVATIONS =  100
THE SAMPLE INPUT VECTORS WILL BE PRINTED ALONG WITH THEIR CORRESPONDING RANKS
HISTOGRAMS OF THE ACTUAL SAMPLE WILL BE PLOTTED FOR EACH INPUT VARIABLE
THE CORRELATION MATRICES (RAW DATA AND RANK CORRELATIONS) WILL BE PRINTED
A PLOT FILE OF SAMPLED VALUES AND CUMULATIVE PROBABILITIES WILL BE CREATED

TITLE SDB Name = INFIL.SDB        , Ver = 1                12/01/99 17:58:05
VARIABLE  DISTRIBUTION          RANGE            LABEL
    1     NORMAL            0.0000E+00 TO   4.0000E-02  FOOTPRNT  BRPOROS
    2     NORMAL             1.000     TO    5.000     FOOTPRNT  BRZDEPTH
    3     NORMAL            0.5000     TO    1.500     FOOTPRNT  SOILDEPM
    4     NORMAL            0.6000     TO    1.400     FOOTPRNT  PRECIPM
    5     NORMAL            0.6000     TO    1.400     FOOTPRNT  POTETMUL
    6     LOGNORMAL         5.0000E-02 TO    20.00     FOOTPRNT  BRPERM
    7     LOGNORMAL         5.0000E-02 TO    20.00     FOOTPRNT  SOILPERM
    8     NORMAL             5.000     TO    15.00     FOOTPRNT  ETCOEFFA
    9     NORMAL            0.5400     TO    1.540     FOOTPRNT  ETCOEFFB
   10     NORMAL            1.0000E-02 TO   0.4900     FOOTPRNT  FLAREA
   11     UNIFORM           0.0000E+00 TO   0.2000     FOOTPRNT  SUBPAR1
```

12 UNIFORM 0.7800 TO 2.780 FOOTPRNT SNOPAR1

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

LATIN HYPERCUBE SAMPLE INPUT VECTORS

RUN NO.	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)	X(8)	X(9)	X(10)
1	2.951E-02	3.747E+00	1.128E+00	1.115E+00	9.015E-01	5.482E+00	2.228E+00	1.061E+01	1.102E+00	3.044E-01
2	1.970E-02	5.000E+00	8.430E-01	8.211E-01	7.525E-01	5.518E-01	3.978E+00	1.185E+01	1.158E+00	1.557E-01
3	2.717E-02	4.735E+00	9.928E-01	9.971E-01	1.077E+00	2.689E-01	1.046E+00	9.708E+00	8.562E-01	3.208E-01
4	1.056E-02	3.048E+00	8.193E-01	9.582E-01	1.375E+00	7.215E+00	1.622E+00	1.180E+01	1.165E+00	4.268E-01
5	2.679E-02	2.762E+00	1.120E+00	1.024E+00	9.981E-01	6.245E+00	1.303E+00	7.633E+00	1.177E+00	2.193E-01
6	2.112E-02	3.101E+00	1.250E+00	6.208E-01	8.708E-01	8.638E-01	2.299E+00	1.045E+01	7.500E-01	1.340E-01
7	2.792E-02	3.020E+00	1.050E+00	9.348E-01	9.136E-01	4.410E-01	4.622E-01	9.903E+00	1.354E+00	3.492E-01
8	2.472E-02	2.557E+00	6.638E-01	1.194E+00	1.020E+00	2.081E-01	1.028E+00	9.261E+00	1.193E+00	1.893E-01
9	1.766E-02	2.406E+00	7.988E-01	1.144E+00	1.104E+00	3.344E+00	7.527E-01	7.565E+00	5.400E-01	8.259E-02
10	1.698E-02	2.726E+00	1.162E+00	1.073E+00	8.152E-01	1.935E+00	1.742E-01	5.707E+00	1.467E+00	2.785E-01
11	1.428E-02	4.024E+00	9.427E-01	1.252E+00	7.342E-01	1.009E+00	2.326E-01	8.754E+00	6.023E-01	1.303E-01
12	1.011E-02	2.613E+00	1.114E+00	8.813E-01	1.123E+00	9.765E+00	8.338E-01	9.673E+00	1.150E+00	2.856E-01
13	2.102E-02	3.365E+00	1.218E+00	1.170E+00	1.135E+00	1.083E+00	7.873E-01	1.351E+01	1.403E+00	4.811E-01
14	1.096E-02	3.413E+00	8.922E-01	8.351E-01	1.099E+00	4.265E+00	1.239E+00	6.349E+00	1.325E+00	2.183E-01
15	4.406E-03	3.695E+00	9.097E-01	8.849E-01	9.188E-01	5.913E-01	4.144E-01	1.071E+01	1.062E+00	2.707E-01
16	1.511E-02	4.786E+00	9.019E-01	1.177E+00	1.049E+00	2.573E+00	2.822E+00	8.652E+00	1.265E+00	2.907E-01
17	3.372E-02	3.874E+00	9.776E-01	1.102E+00	1.006E+00	3.734E+00	1.302E-01	8.971E+00	9.041E-01	4.194E-01
18	2.601E-02	3.684E+00	1.184E+00	6.838E-01	9.656E-01	1.484E+01	4.255E+00	6.677E+00	6.412E-01	3.573E-01
19	8.194E-03	4.063E+00	7.482E-01	9.258E-01	9.488E-01	2.000E+01	3.202E-01	1.108E+01	9.933E-01	1.944E-01
20	3.486E-02	2.895E+00	1.403E+00	1.153E+00	9.650E-01	3.490E+00	1.905E-01	1.452E+01	6.666E-01	1.836E-01
21	3.307E-02	2.450E+00	9.507E-01	9.036E-01	1.206E+00	5.304E-01	9.365E-01	1.106E+01	1.026E+00	2.750E-01
22	1.842E-02	2.987E+00	7.289E-01	1.225E+00	8.440E-01	1.331E+00	1.984E+00	9.511E+00	1.120E+00	2.346E-01

23	3.768E-02	3.449E+00	1.067E+00	9.242E-01	1.038E+00	1.627E+00	3.150E+00	1.052E+01	1.312E+00	1.000E-02
24	2.136E-02	3.305E+00	1.020E+00	9.756E-01	1.072E+00	2.208E-01	2.000E+01	1.500E+01	8.043E-01	2.331E-01
25	1.748E-02	1.707E+00	8.339E-01	1.357E+00	8.334E-01	4.768E+00	5.215E-01	1.330E+01	7.821E-01	1.279E-01
26	2.218E-02	3.462E+00	1.030E+00	8.575E-01	8.780E-01	1.739E-01	7.220E+00	8.828E+00	9.812E-01	2.615E-01
27	1.115E-02	3.956E+00	6.937E-01	9.433E-01	6.727E-01	6.810E-01	1.391E+00	1.092E+01	1.143E+00	2.931E-01
28	1.301E-02	2.738E+00	5.703E-01	9.059E-01	1.024E+00	1.121E+00	3.548E+00	1.289E+01	1.049E+00	3.741E-01
29	1.906E-02	1.907E+00	7.107E-01	9.912E-01	7.209E-01	1.683E+00	6.061E-01	9.762E+00	7.139E-01	4.411E-01
30	2.496E-02	1.787E+00	1.036E+00	1.038E+00	1.082E+00	2.466E+00	2.170E+00	1.009E+01	1.052E+00	3.320E-01
31	2.253E-02	4.295E+00	5.545E-01	7.330E-01	1.058E+00	8.050E-01	9.500E-01	9.890E+00	8.274E-01	1.745E-01
32	2.574E-02	3.758E+00	1.112E+00	1.325E+00	9.705E-01	1.241E+00	1.643E+00	7.338E+00	9.189E-01	3.029E-01
33	2.626E-02	3.071E+00	7.415E-01	1.070E+00	1.028E+00	4.637E-01	8.584E-01	6.964E+00	8.921E-01	3.356E-01
34	5.092E-03	2.643E+00	1.230E+00	1.282E+00	9.279E-01	2.759E+00	5.397E-01	8.395E+00	9.121E-01	7.665E-02
35	8.502E-03	3.510E+00	8.561E-01	8.227E-01	9.762E-01	4.007E+00	1.055E+01	1.207E+01	9.748E-01	3.830E-01
36	2.156E-02	2.818E+00	1.025E+00	1.034E+00	8.252E-01	9.187E+00	4.586E+00	1.256E+01	1.022E+00	1.685E-01
37	3.114E-02	3.558E+00	8.179E-01	7.822E-01	8.854E-01	8.770E-02	7.420E-01	1.142E+01	1.220E+00	2.143E-01
38	2.437E-02	1.977E+00	8.099E-01	1.013E+00	1.283E+00	3.189E+00	1.544E+00	7.880E+00	9.466E-01	2.052E-01
39	2.371E-02	2.787E+00	7.816E-01	1.130E+00	1.153E+00	5.839E-01	2.642E-01	7.981E+00	7.724E-01	2.544E-01
40	1.952E-02	4.004E+00	1.191E+00	1.118E+00	1.042E+00	2.911E+00	1.264E+00	1.268E+01	9.624E-01	3.420E-01
41	1.344E-02	1.187E+00	1.500E+00	8.661E-01	1.014E+00	1.432E+00	6.631E-01	1.302E+01	8.986E-01	1.622E-01
42	2.832E-02	2.597E+00	8.675E-01	1.020E+00	1.120E+00	2.417E-01	1.702E+00	1.072E+01	1.185E+00	2.517E-01
43	2.869E-02	2.947E+00	1.015E+00	9.552E-01	6.915E-01	1.170E+01	4.803E-01	1.119E+01	1.106E+00	2.251E-01
44	3.074E-02	3.347E+00	1.099E+00	1.262E+00	1.033E+00	2.819E-01	2.454E+00	1.391E+01	1.006E+00	3.529E-01
45	3.607E-02	3.278E+00	1.271E+00	1.004E+00	1.112E+00	7.296E-01	2.139E-01	8.605E+00	1.298E+00	1.457E-01
46	9.806E-03	1.000E+00	8.868E-01	1.008E+00	6.082E-01	8.379E-01	1.749E+01	8.050E+00	1.093E+00	2.392E-01
47	1.866E-02	2.326E+00	9.556E-01	1.108E+00	8.830E-01	5.019E-01	3.003E+00	1.238E+01	1.032E+00	1.002E-01

48	1.232E-02	2.423E+00	1.242E+00	1.054E+00	1.066E+00	1.851E-01	1.186E+01	9.090E+00	1.179E+00	4.109E-02
49	2.650E-02	2.647E+00	6.129E-01	8.403E-01	1.011E+00	1.750E+00	2.478E-01	1.026E+01	8.205E-01	3.896E-01
50	1.980E-02	2.000E+00	8.696E-01	7.585E-01	8.964E-01	3.005E-01	3.963E-01	9.394E+00	1.365E+00	1.515E-01
51	-3.725E-09	2.494E+00	6.738E-01	1.084E+00	1.212E+00	1.986E+00	6.346E-02	1.129E+01	9.522E-01	2.721E-01
52	1.660E-02	2.876E+00	1.203E+00	8.053E-01	1.143E+00	7.108E-01	5.560E-01	1.095E+01	1.016E+00	3.688E-01
53	1.487E-02	3.232E+00	9.316E-01	1.080E+00	1.323E+00	6.212E-01	1.783E+00	9.294E+00	1.490E+00	2.558E-01
54	2.992E-02	2.211E+00	9.139E-01	7.558E-01	7.670E-01	1.169E+00	9.037E-01	8.741E+00	1.385E+00	2.886E-01
55	3.174E-02	2.228E+00	1.071E+00	9.101E-01	9.056E-01	6.346E-01	7.149E-01	1.016E+01	1.224E+00	2.088E-01
56	3.032E-02	3.119E+00	1.153E+00	1.125E+00	8.406E-01	8.091E+00	9.303E+00	6.063E+00	9.879E-01	3.480E-01
57	1.821E-02	2.166E+00	1.093E+00	9.619E-01	1.110E+00	8.911E-01	2.900E-01	9.610E+00	1.131E+00	1.902E-01
58	2.900E-02	4.364E+00	7.745E-01	1.051E+00	9.838E-01	7.533E-01	3.644E+00	5.379E+00	1.281E+00	2.213E-01
59	2.669E-03	1.813E+00	1.339E+00	9.517E-01	9.823E-01	1.988E-01	5.110E+00	8.417E+00	1.088E+00	3.006E-01
60	2.287E-02	3.903E+00	6.246E-01	8.438E-01	1.244E+00	2.347E+00	6.503E-01	1.081E+01	8.598E-01	1.142E-01
61	1.927E-02	1.890E+00	1.418E+00	1.099E+00	9.216E-01	7.101E-02	1.851E+00	1.042E+01	9.649E-01	2.669E-01
62	2.038E-02	2.126E+00	1.176E+00	6.000E-01	9.348E-01	1.504E+00	5.000E-02	7.819E+00	9.996E-01	2.806E-01
63	7.731E-03	3.803E+00	1.039E+00	8.120E-01	7.936E-01	4.195E-01	3.768E-01	1.230E+01	1.125E+00	3.984E-01
64	6.695E-03	4.216E+00	1.138E+00	6.596E-01	1.349E+00	1.261E+00	1.933E+00	7.715E+00	1.167E+00	6.914E-02
65	1.400E-02	2.265E+00	8.411E-01	1.216E+00	1.267E+00	3.142E-01	1.205E+00	1.215E+01	1.099E+00	2.014E-01
66	2.317E-02	3.605E+00	1.004E+00	9.833E-01	1.163E+00	2.701E+00	5.684E-01	9.820E+00	1.329E+00	3.125E-01
67	1.559E-02	3.379E+00	1.358E+00	9.149E-01	6.000E-01	1.808E+00	1.454E+00	1.155E+01	8.758E-01	3.251E-01
68	1.172E-02	3.493E+00	1.283E+00	1.160E+00	1.070E+00	1.435E-01	2.537E+00	1.160E+01	1.060E+00	4.458E-01
69	1.328E-02	2.374E+00	1.140E+00	7.244E-01	7.780E-01	1.360E+00	1.375E-01	1.205E+01	1.140E+00	9.449E-02
70	6.542E-03	3.031E+00	7.206E-01	1.043E+00	9.298E-01	2.525E-01	8.168E-01	8.219E+00	9.714E-01	3.080E-01
71	1.423E-03	3.167E+00	1.080E+00	1.146E+00	1.147E+00	1.007E-01	7.752E-02	1.011E+01	1.233E+00	2.603E-01
72	3.223E-02	3.311E+00	1.255E+00	9.393E-01	1.089E+00	1.859E+00	2.096E+00	7.045E+00	7.620E-01	2.440E-01
73	3.729E-02	3.214E+00	7.627E-01	8.919E-01	8.610E-01	5.155E-01	6.235E-01	1.149E+01	1.078E+00	1.793E-01

74	2.285E-02	4.201E+00	1.167E+00	1.195E+00	9.398E-01	3.371E-01	1.073E+00	8.319E+00	1.010E+00	3.810E-01
75	1.580E-02	2.921E+00	7.914E-01	1.064E+00	8.032E-01	9.675E-01	7.436E+00	1.084E+01	1.114E+00	1.393E-01
76	2.080E-02	3.189E+00	1.315E+00	7.045E-01	1.055E+00	6.645E-01	4.461E-01	1.171E+01	1.043E+00	2.828E-01
77	2.017E-02	4.565E+00	1.289E+00	1.400E+00	1.266E+00	1.569E+00	1.382E+00	1.055E+01	9.288E-01	1.545E-01
78	2.540E-02	2.704E+00	1.063E+00	1.088E+00	9.907E-01	9.291E-01	9.998E-01	9.445E+00	1.540E+00	1.066E-01
79	1.802E-02	2.056E+00	8.807E-01	1.315E+00	9.447E-01	7.770E-01	2.611E+00	6.548E+00	1.081E+00	2.027E-01
80	1.718E-02	3.536E+00	9.604E-01	1.234E+00	8.629E-01	2.049E+00	1.139E+00	1.119E+01	9.382E-01	1.967E-01
81	2.780E-02	3.151E+00	9.188E-01	8.981E-01	1.400E+00	4.264E-01	4.264E-01	1.028E+01	8.511E-01	3.135E-01
82	1.669E-02	2.857E+00	1.060E+00	9.919E-01	1.167E+00	1.139E-01	2.621E-01	8.532E+00	7.362E-01	2.267E-01
83	2.750E-02	2.076E+00	9.952E-01	1.208E+00	1.130E+00	4.880E+00	9.401E-02	1.245E+01	1.256E+00	1.205E-01
84	1.525E-02	4.414E+00	1.008E+00	1.092E+00	8.092E-01	1.307E-01	1.117E+00	8.147E+00	1.037E+00	1.641E-01
85	2.180E-02	2.977E+00	9.365E-01	1.183E+00	1.180E+00	2.197E+00	3.311E+00	1.374E+01	1.418E+00	2.969E-01
86	2.383E-02	2.299E+00	9.788E-01	8.553E-01	8.915E-01	3.010E+00	3.362E-01	1.037E+01	1.213E+00	3.386E-01
87	4.000E-02	1.424E+00	9.638E-01	7.756E-01	1.230E+00	5.000E-02	1.508E+00	1.135E+01	9.410E-01	1.839E-01
88	1.368E-02	3.619E+00	1.105E+00	9.654E-01	1.195E+00	5.951E+00	3.633E-01	1.280E+01	1.252E+00	2.421E-01
89	1.257E-02	3.640E+00	1.320E+00	8.757E-01	9.562E-01	4.796E-01	3.537E-01	1.001E+01	9.270E-01	2.452E-01
90	1.455E-02	2.548E+00	1.085E+00	9.305E-01	1.233E+00	9.904E-01	4.531E+00	8.905E+00	8.151E-01	2.635E-01
91	2.506E-02	2.362E+00	5.000E-01	1.044E+00	1.094E+00	3.620E-01	2.940E-01	7.136E+00	7.918E-01	3.263E-01
92	1.871E-02	1.255E+00	1.146E+00	9.818E-01	1.189E+00	1.477E+00	6.268E+00	9.537E+00	8.358E-01	4.900E-01
93	2.355E-02	3.848E+00	9.887E-01	1.138E+00	7.849E-01	1.579E-01	1.578E-01	1.409E+01	6.957E-01	2.498E-01
94	9.140E-03	3.261E+00	1.049E+00	9.710E-01	1.047E+00	1.044E+00	6.952E-01	9.954E+00	1.287E+00	2.111E-01
95	2.054E-02	2.844E+00	9.269E-01	1.014E+00	9.574E-01	2.279E+00	5.026E-01	1.194E+01	1.199E+00	1.704E-01
96	1.629E-02	4.123E+00	1.466E+00	1.060E+00	1.002E+00	3.931E-01	1.123E-01	5.000E+00	8.723E-01	2.310E-01
97	1.603E-02	2.688E+00	9.701E-01	7.967E-01	8.518E-01	3.779E-01	2.902E+00	9.016E+00	8.878E-01	3.641E-01
98	1.211E-02	2.519E+00	8.959E-01	8.728E-01	1.175E+00	1.182E+00	5.797E+00	9.137E+00	5.755E-01	3.582E-02

99	2.408E-02	1.588E+00	8.531E-01	1.290E+00	7.100E-01	3.385E-01	2.017E-01	9.210E+00	1.241E+00	4.031E-01
100	2.205E-02	1.569E+00	1.206E+00	1.028E+00	9.944E-01	4.535E+00	5.309E+00	7.390E+00	1.072E+00	3.177E-01

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

LATIN HYPERCUBE SAMPLE INPUT VECTORS

RUN NO.	X(11)	X(12)
1	1.571E-01	2.158E+00
2	7.899E-02	1.084E+00
3	4.278E-02	1.073E+00
4	1.945E-01	1.778E+00
5	2.968E-04	8.480E-01
6	1.137E-01	2.000E+00
7	4.974E-02	1.687E+00
8	1.170E-01	1.806E+00
9	1.689E-01	1.169E+00
10	1.595E-01	1.047E+00
11	1.192E-01	2.216E+00
12	1.819E-01	2.654E+00
13	1.973E-01	1.849E+00
14	6.094E-02	1.668E+00
15	2.240E-02	1.587E+00
16	1.261E-01	2.285E+00
17	1.061E-01	8.812E-01
18	3.334E-02	2.513E+00
19	1.374E-01	1.399E+00
20	1.318E-01	2.001E+00
21	8.391E-02	1.916E+00

22	5.225E-03	1.629E+00
23	1.150E-01	1.945E+00
24	7.343E-02	1.821E+00
25	1.624E-01	2.581E+00
26	5.193E-02	1.747E+00
27	8.031E-02	2.772E+00
28	7.456E-02	1.787E+00
29	1.762E-02	9.068E-01
30	7.349E-03	9.300E-01
31	1.549E-01	8.084E-01
32	1.339E-01	2.498E+00
33	1.357E-01	2.758E+00
34	4.017E-02	1.710E+00
35	8.800E-02	2.732E+00
36	1.443E-01	2.047E+00
37	1.741E-01	2.604E+00
38	8.808E-02	2.339E+00
39	9.344E-02	1.139E+00
40	4.684E-02	1.110E+00
41	1.391E-02	2.166E+00
42	1.611E-01	1.872E+00
43	1.720E-01	8.214E-01
44	1.468E-02	2.232E+00
45	9.658E-02	2.523E+00
46	1.760E-01	1.498E+00

47	1.105E-01	1.577E+00
48	1.425E-01	1.005E+00
49	1.491E-01	1.333E+00
50	6.248E-02	2.022E+00
51	2.432E-02	2.469E+00
52	1.255E-01	9.749E-01
53	9.936E-02	1.306E+00
54	1.405E-01	1.272E+00
55	8.284E-03	2.407E+00
56	5.477E-02	2.197E+00
57	3.717E-02	1.231E+00
58	1.097E-01	2.430E+00
59	8.425E-02	1.030E+00
60	1.711E-01	2.663E+00
61	1.856E-01	2.709E+00
62	1.027E-01	2.255E+00
63	1.175E-02	2.094E+00
64	5.221E-02	1.509E+00
65	3.294E-03	2.108E+00
66	6.838E-02	2.632E+00
67	3.874E-02	1.723E+00
68	3.449E-02	1.430E+00
69	2.707E-02	2.273E+00
70	9.583E-02	1.477E+00
71	7.627E-02	1.368E+00
72	4.562E-02	1.199E+00

73	5.641E-02	9.557E-01
74	1.641E-01	1.211E+00
75	3.159E-02	2.301E+00
76	1.517E-01	2.399E+00
77	1.051E-01	1.257E+00
78	1.531E-01	1.538E+00
79	1.675E-01	1.407E+00
80	1.985E-02	1.459E+00
81	7.060E-02	1.617E+00
82	6.630E-02	2.065E+00
83	2.975E-02	1.152E+00
84	5.834E-02	2.578E+00
85	1.010E-01	1.293E+00
86	1.391E-01	1.921E+00
87	1.292E-01	1.644E+00
88	1.214E-01	9.884E-01
89	1.992E-01	1.971E+00
90	1.928E-01	2.556E+00
91	2.063E-02	2.458E+00
92	6.468E-02	2.350E+00
93	1.906E-01	1.356E+00
94	1.783E-01	1.548E+00
95	1.475E-01	2.690E+00
96	9.158E-02	2.367E+00
97	1.869E-01	7.898E-01

98	1.898E-01	8.639E-01
99	1.828E-01	1.880E+00
100	1.222E-01	2.122E+00

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

RANKS OF LATIN HYPERCUBE SAMPLE INPUT VECTORS

RUN NO.	X(1)	X(2)	X(3)	X(4)	X(5)	X(6)	X(7)	X(8)	X(9)	X(10)
1	87.	81.	73.	75.	29.	91.	74.	62.	62.	71.
2	49.	100.	24.	15.	8.	33.	86.	81.	71.	19.
3	80.	98.	49.	50.	68.	16.	52.	45.	20.	76.
4	14.	53.	21.	41.	99.	94.	65.	80.	72.	96.
5	79.	40.	72.	56.	50.	93.	59.	14.	74.	39.
6	56.	55.	88.	2.	23.	46.	75.	59.	9.	14.
7	83.	51.	60.	36.	31.	27.	28.	49.	93.	84.
8	71.	31.	6.	87.	55.	12.	51.	37.	77.	28.
9	40.	25.	18.	80.	73.	83.	42.	13.	1.	6.
10	37.	38.	78.	67.	15.	70.	9.	3.	98.	61.
11	26.	89.	40.	93.	7.	51.	13.	29.	3.	13.
12	13.	33.	71.	25.	77.	97.	45.	44.	70.	64.
13	55.	67.	85.	84.	79.	53.	43.	95.	96.	99.
14	15.	69.	31.	17.	72.	87.	57.	5.	91.	38.
15	4.	80.	34.	26.	32.	35.	25.	63.	55.	58.
16	29.	99.	33.	85.	62.	77.	79.	27.	86.	66.
17	95.	85.	46.	73.	52.	85.	6.	32.	27.	95.
18	76.	79.	81.	4.	43.	99.	87.	7.	4.	86.
19	9.	90.	13.	34.	39.	100.	19.	70.	42.	30.
20	96.	46.	97.	82.	42.	84.	10.	99.	5.	26.

21	94.	27.	41.	29.	89.	32.	48.	69.	48.	60.
22	43.	50.	11.	91.	19.	59.	71.	41.	65.	45.
23	99.	70.	63.	33.	59.	65.	82.	60.	90.	1.
24	57.	64.	54.	45.	67.	13.	100.	100.	14.	44.
25	39.	7.	22.	99.	17.	89.	31.	94.	12.	12.
26	61.	71.	56.	21.	24.	9.	94.	30.	40.	55.
27	16.	87.	8.	38.	3.	39.	61.	67.	69.	67.
28	21.	39.	3.	30.	56.	54.	84.	92.	52.	89.
29	46.	11.	9.	48.	6.	66.	35.	46.	7.	97.
30	72.	8.	57.	59.	69.	76.	73.	52.	53.	79.
31	62.	94.	2.	7.	64.	44.	49.	48.	17.	24.
32	75.	82.	70.	98.	44.	57.	66.	11.	29.	70.
33	77.	54.	12.	66.	57.	28.	46.	8.	25.	80.
34	5.	34.	86.	95.	34.	79.	32.	23.	28.	5.
35	10.	73.	26.	16.	45.	86.	97.	84.	39.	91.
36	58.	42.	55.	58.	16.	96.	89.	89.	47.	22.
37	91.	75.	20.	11.	26.	3.	41.	75.	80.	37.
38	70.	12.	19.	53.	96.	82.	64.	17.	34.	34.
39	67.	41.	16.	78.	82.	34.	16.	18.	11.	52.
40	48.	88.	82.	76.	60.	80.	58.	90.	36.	82.
41	23.	2.	100.	22.	54.	61.	38.	93.	26.	20.
42	84.	32.	27.	55.	76.	14.	67.	64.	76.	51.
43	85.	48.	53.	40.	4.	98.	29.	71.	63.	41.
44	90.	66.	68.	94.	58.	17.	76.	97.	44.	85.
45	97.	63.	90.	51.	75.	41.	12.	26.	89.	16.

46	12.	1.	30.	52.	2.	45.	99.	19.	60.	46.
47	44.	22.	42.	74.	25.	30.	81.	87.	49.	8.
48	19.	26.	87.	63.	65.	10.	98.	34.	75.	3.
49	78.	35.	4.	18.	53.	67.	14.	55.	16.	92.
50	50.	13.	28.	9.	28.	18.	24.	39.	94.	17.
51	1.	28.	7.	69.	90.	71.	2.	73.	35.	59.
52	35.	45.	83.	13.	80.	40.	33.	68.	46.	88.
53	28.	61.	38.	68.	97.	36.	68.	38.	99.	53.
54	88.	18.	35.	8.	9.	55.	47.	28.	95.	65.
55	92.	19.	64.	31.	30.	37.	40.	54.	81.	35.
56	89.	56.	77.	77.	18.	95.	96.	4.	41.	83.
57	42.	17.	67.	42.	74.	47.	17.	43.	67.	29.
58	86.	95.	15.	62.	47.	42.	85.	2.	87.	40.
59	3.	9.	95.	39.	46.	11.	90.	24.	59.	69.
60	64.	86.	5.	19.	93.	75.	37.	65.	21.	10.
61	47.	10.	98.	72.	33.	2.	69.	58.	37.	57.
62	52.	16.	80.	1.	36.	63.	1.	16.	43.	62.
63	8.	83.	58.	14.	12.	25.	23.	86.	66.	93.
64	7.	93.	74.	3.	98.	58.	70.	15.	73.	4.
65	25.	20.	23.	90.	95.	19.	56.	85.	61.	32.
66	65.	76.	51.	47.	83.	78.	34.	47.	92.	73.
67	31.	68.	96.	32.	1.	68.	62.	77.	23.	77.
68	17.	72.	91.	83.	66.	7.	77.	78.	54.	98.
69	22.	24.	75.	6.	10.	60.	7.	83.	68.	7.
70	6.	52.	10.	60.	35.	15.	44.	21.	38.	72.
71	2.	58.	65.	81.	81.	4.	3.	53.	82.	54.

72	93.	65.	89.	37.	70.	69.	72.	9.	10.	48.
73	98.	60.	14.	27.	21.	31.	36.	76.	57.	25.
74	63.	92.	79.	88.	37.	20.	53.	22.	45.	90.
75	32.	47.	17.	65.	13.	49.	95.	66.	64.	15.
76	54.	59.	93.	5.	63.	38.	27.	79.	51.	63.
77	51.	97.	92.	100.	94.	64.	60.	61.	31.	18.
78	74.	37.	62.	70.	48.	48.	50.	40.	100.	9.
79	41.	14.	29.	97.	38.	43.	78.	6.	58.	33.
80	38.	74.	43.	92.	22.	72.	55.	72.	32.	31.
81	82.	57.	36.	28.	100.	26.	26.	56.	19.	74.
82	36.	44.	61.	49.	84.	5.	15.	25.	8.	42.
83	81.	15.	50.	89.	78.	90.	4.	88.	85.	11.
84	30.	96.	52.	71.	14.	6.	54.	20.	50.	21.
85	59.	49.	39.	86.	86.	73.	83.	96.	97.	68.
86	68.	21.	47.	20.	27.	81.	20.	57.	79.	81.
87	100.	4.	44.	10.	91.	1.	63.	74.	33.	27.
88	24.	77.	69.	43.	88.	92.	22.	91.	84.	47.
89	20.	78.	94.	24.	40.	29.	21.	51.	30.	49.
90	27.	30.	66.	35.	92.	50.	88.	31.	15.	56.
91	73.	23.	1.	61.	71.	22.	18.	10.	13.	78.
92	45.	3.	76.	46.	87.	62.	93.	42.	18.	100.
93	66.	84.	48.	79.	11.	8.	8.	98.	6.	50.
94	11.	62.	59.	44.	61.	52.	39.	50.	88.	36.
95	53.	43.	37.	54.	41.	74.	30.	82.	78.	23.
96	34.	91.	99.	64.	51.	24.	5.	1.	22.	43.

97	33.	36.	45.	12.	20.	23.	80.	33.	24.	87.
98	18.	29.	32.	23.	85.	56.	92.	35.	2.	2.
99	69.	6.	25.	96.	5.	21.	11.	36.	83.	94.
100	60.	5.	84.	57.	49.	88.	91.	12.	56.	75.

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

RANKS OF LATIN HYPERCUBE SAMPLE INPUT VECTORS

RUN NO.	X(11)	X(12)
1	79.	69.
2	40.	16.
3	22.	15.
4	98.	50.
5	1.	4.
6	57.	61.
7	25.	46.
8	59.	52.
9	85.	20.
10	80.	14.
11	60.	72.
12	91.	94.
13	99.	54.
14	31.	45.
15	12.	41.
16	64.	76.
17	54.	6.
18	17.	87.
19	69.	31.

20	66.	62.
21	42.	57.
22	3.	43.
23	58.	59.
24	37.	53.
25	82.	91.
26	26.	49.
27	41.	100.
28	38.	51.
29	9.	7.
30	4.	8.
31	78.	2.
32	67.	86.
33	68.	99.
34	21.	47.
35	44.	98.
36	73.	64.
37	88.	92.
38	45.	78.
39	47.	18.
40	24.	17.
41	7.	70.
42	81.	55.
43	87.	3.
44	8.	73.

45	49.	88.
46	89.	36.
47	56.	40.
48	72.	12.
49	75.	28.
50	32.	63.
51	13.	85.
52	63.	10.
53	50.	27.
54	71.	25.
55	5.	82.
56	28.	71.
57	19.	23.
58	55.	83.
59	43.	13.
60	86.	95.
61	93.	97.
62	52.	74.
63	6.	66.
64	27.	37.
65	2.	67.
66	35.	93.
67	20.	48.
68	18.	33.
69	14.	75.
70	48.	35.

71	39.	30.
72	23.	21.
73	29.	9.
74	83.	22.
75	16.	77.
76	76.	81.
77	53.	24.
78	77.	38.
79	84.	32.
80	10.	34.
81	36.	42.
82	34.	65.
83	15.	19.
84	30.	90.
85	51.	26.
86	70.	58.
87	65.	44.
88	61.	11.
89	100.	60.
90	97.	89.
91	11.	84.
92	33.	79.
93	96.	29.
94	90.	39.
95	74.	96.

96	46.	80.
97	94.	1.
98	95.	5.
99	92.	56.
100	62.	68.

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 1 NORMAL DISTRIBUTION

MIDPOINT	FREQ.	
-0.1000000E-02	1	X
0.1000000E-02	1	X
0.3000000E-02	1	X
0.5000001E-02	2	XX
0.7000001E-02	3	XXX
0.9000001E-02	4	XXXX
0.1100000E-01	5	XXXXX
0.1300000E-01	7	XXXXXXX
0.1500000E-01	8	XXXXXXXX
0.1700000E-01	8	XXXXXXXX
0.1900000E-01	10	XXXXXXXXXX
0.2100000E-01	9	XXXXXXXXX
0.2300000E-01	9	XXXXXXXXX
0.2500000E-01	7	XXXXXXX
0.2700000E-01	8	XXXXXXXX
0.2900000E-01	5	XXXXX
0.3100001E-01	4	XXXX
0.3300001E-01	3	XXX
0.3500001E-01	1	X
0.3700001E-01	3	XXX
0.3900001E-01	1	X
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
-0.3725290E-08	0.4000000E-01	0.4000001E-01	0.1998924E-01	0.1998468E-01	0.7142533E-04

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 2 NORMAL DISTRIBUTION

MIDPOINT	FREQ.	
0.9000002	1	X
1.100000	1	X
1.300000	1	X
1.500000	3	XXX
1.700000	2	XX
1.900000	5	XXXXX
2.100000	4	XXXX
2.300000	7	XXXXXXX
2.500000	8	XXXXXXXX
2.700001	9	XXXXXXXXX
2.900001	9	XXXXXXXXX
3.100001	9	XXXXXXXXX
3.300001	9	XXXXXXXXX
3.500001	7	XXXXXXX
3.700001	7	XXXXXXX
3.900001	5	XXXXX
4.100001	4	XXXX
4.300001	4	XXXX
4.500000	2	XX
4.700000	2	XX
4.900000	0	
5.100000	1	X
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.9999995	5.000000	4.000001	2.999520	3.003353	0.7086340

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 3 NORMAL DISTRIBUTION

MIDPOINT	FREQ.	
0.5249998	1	X
0.5749998	2	XX
0.6249998	2	XX
0.6749998	3	XXX
0.7249998	5	XXXXX
0.7749999	5	XXXXX
0.8249999	6	XXXXXX
0.8749999	8	XXXXXXXX
0.9249999	8	XXXXXXXX
0.9749999	10	XXXXXXXXXX
1.025000	9	XXXXXXXXX
1.075000	9	XXXXXXXXX

1.125000	8	XXXXXXXX
1.175000	6	XXXXXX
1.225000	5	XXXXX
1.275000	5	XXXXX
1.325000	3	XXX
1.375000	1	X
1.424999	2	XX
1.474999	1	X
1.524999	1	X
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.4999999	1.500000	1.000000	1.000313	0.9995143	0.4426514E-01

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 4 NORMAL DISTRIBUTION

MIDPOINT	FREQ.	
0.6199999	2	XX
0.6599999	1	X
0.6999999	2	XX
0.7399999	4	XXXX
0.7800000	3	XXX
0.8200000	5	XXXXX
0.8600000	7	XXXXXXX
0.9000000	8	XXXXXXXX
0.9400001	9	XXXXXXXXX
0.9800001	9	XXXXXXXXX
1.020000	9	XXXXXXXXX
1.060000	8	XXXXXXXX
1.100000	9	XXXXXXXXX
1.140000	7	XXXXXXX
1.180000	5	XXXXX
1.220000	4	XXXX
1.260000	2	XX
1.300000	3	XXX
1.340000	2	XX
1.380000	0	
1.420000	1	X
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE

0.6000000	1.400000	0.8000001	0.9998415	1.000569	0.2853096E-01

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 5 NORMAL DISTRIBUTION

MIDPOINT	FREQ.	
0.6199999	2	XX
0.6599999	1	X
0.6999999	2	XX
0.7399999	3	XXX
0.7800000	4	XXXX
0.8200000	5	XXXXX
0.8600000	7	XXXXXXX
0.9000000	8	XXXXXXXX
0.9400001	9	XXXXXXXXX
0.9800001	9	XXXXXXXXX
1.020000	9	XXXXXXXXX
1.060000	9	XXXXXXXXX
1.100000	7	XXXXXXX
1.140000	7	XXXXXXX
1.180000	6	XXXXXX
1.220000	4	XXXX
1.260000	3	XXX
1.300000	1	X
1.340000	2	XX
1.380000	1	X
1.420000	1	X
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.6000000	1.400000	0.8000001	1.000062	0.9998701	0.2863144E-01

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 6 LOGNORMAL DISTRIBUTION

MIDPOINT	FREQ.	
0.4999997	50	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1.499999	21	XXXXXXXXXXXXXXXXXXXXX
2.499998	9	XXXXXXXXX
3.499998	5	XXXXX
4.499997	5	XXXXX
5.499997	2	XX

6.499996	1	X
7.499996	1	X
8.499995	1	X
9.499994	2	XX
10.49999	0	
11.49999	1	X
12.49999	0	
13.49999	0	
14.49999	1	X
15.49999	0	
16.49999	0	
17.49999	0	
18.49999	0	
19.49999	0	
20.49999	1	X
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.4999997E-01	20.00001	19.95001	2.130778	0.9996805	9.782403

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 7 LOGNORMAL DISTRIBUTION

MIDPOINT	FREQ.	
0.4999997	50	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1.499999	21	XXXXXXXXXXXXXXXXXXXXX
2.499998	9	XXXXXXXXX
3.499998	6	XXXXXX
4.499997	3	XXX
5.499997	3	XXX
6.499996	1	X
7.499996	2	XX
8.499995	0	
9.499994	1	X
10.49999	1	X
11.49999	1	X
12.49999	0	
13.49999	0	
14.49999	0	
15.49999	0	
16.49999	0	
17.49999	1	X
18.49999	0	
19.49999	0	
20.49999	1	X

100

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.4999997E-01	20.00001	19.95001	2.154755	1.013921	10.60305

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 8 NORMAL DISTRIBUTION

MIDPOINT	FREQ.	
5.249998	2	XX
5.749998	1	X
6.249998	2	XX
6.749998	3	XXX
7.249998	4	XXXX
7.749998	6	XXXXXX
8.249998	6	XXXXXX
8.749998	8	XXXXXXXX
9.249998	8	XXXXXXXX
9.749998	10	XXXXXXXXXX
10.25000	9	XXXXXXXXX
10.75000	9	XXXXXXXXX
11.25000	8	XXXXXXXX
11.75000	6	XXXXXX
12.25000	6	XXXXXX
12.75000	4	XXXX
13.25000	2	XX
13.75000	3	XXX
14.25000	1	X
14.75000	1	X
15.25000	1	X

100

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
4.999999	15.00000	10.00000	9.996411	9.983184	4.441211

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 9 NORMAL DISTRIBUTION

MIDPOINT FREQ.

```
0.5249998          1    X
0.5749998          1    X
0.6249998          2    XX
0.6749998          2    XX
0.7249998          2    XX
0.7749999          5    XXXXX
0.8249999          5    XXXXX
0.8749999          8    XXXXXXXX
0.9249999          8    XXXXXXXX
0.9749999          9    XXXXXXXXX
 1.025000          9    XXXXXXXXX
 1.075000          9    XXXXXXXXX
 1.125000          8    XXXXXXXX
 1.175000          9    XXXXXXXXX
 1.225000          5    XXXXX
 1.275000          6    XXXXXX
 1.325000          3    XXX
 1.375000          3    XXX
 1.424999          2    XX
 1.474999          2    XX
 1.524999          1    X

                  100
```

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.5399998	1.540000	1.000000	1.039693	1.040313	0.4451736E-01

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 10 NORMAL DISTRIBUTION

```
  MIDPOINT          FREQ.

0.1200000E-01         1    X
0.3600001E-01         2    XX
0.6000002E-01         1    X
0.8400002E-01         3    XXX
0.1080000             3    XXX
0.1320000             5    XXXXX
0.1560000             6    XXXXXX
0.1800000             8    XXXXXXXX
0.2040000             8    XXXXXXXX
0.2280000             9    XXXXXXXXX
0.2520001            10    XXXXXXXXXX
0.2760001             8    XXXXXXXX
0.3000001             8    XXXXXXXX
0.3240001             8    XXXXXXXX
0.3480001             6    XXXXXX
```

0.3720002	5	XXXXX
0.3960002	3	XXX
0.4200002	2	XX
0.4440002	2	XX
0.4680002	0	
0.4920003	2	XX
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.9999931E-02	0.4900001	0.4800001	0.2500286	0.2507621	0.1025054E-01

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 11 UNIFORM DISTRIBUTION

MIDPOINT	FREQ.	
0.4949997E-02	5	XXXXX
0.1484999E-01	4	XXXX
0.2474998E-01	5	XXXXX
0.3464998E-01	6	XXXXXX
0.4454997E-01	4	XXXX
0.5444996E-01	6	XXXXXX
0.6434996E-01	5	XXXXX
0.7424995E-01	5	XXXXX
0.8414995E-01	5	XXXXX
0.9404995E-01	4	XXXX
0.1039499	5	XXXXX
0.1138499	5	XXXXX
0.1237499	5	XXXXX
0.1336499	5	XXXXX
0.1435499	5	XXXXX
0.1534499	5	XXXXX
0.1633499	5	XXXXX
0.1732499	5	XXXXX
0.1831499	5	XXXXX
0.1930499	5	XXXXX
0.2029499	1	X
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.2967655E-03	0.1992379	0.1989412	0.9998050E-01	0.1001890	0.3331727E-02

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

HISTOGRAM FOR VARIABLE NO. 12 UNIFORM DISTRIBUTION

MIDPOINT	FREQ.	
0.7424996	1	X
0.8414995	5	XXXXX
0.9404994	5	XXXXX
1.039499	5	XXXXX
1.138499	4	XXXX
1.237499	5	XXXXX
1.336499	5	XXXXX
1.435499	5	XXXXX
1.534499	5	XXXXX
1.633499	5	XXXXX
1.732499	5	XXXXX
1.831499	6	XXXXXX
1.930499	4	XXXX
2.029499	5	XXXXX
2.128499	5	XXXXX
2.227499	5	XXXXX
2.326499	5	XXXXX
2.425499	5	XXXXX
2.524499	4	XXXX
2.623499	6	XXXXXX
2.722499	5	XXXXX
	100	

MIN	MAX	RANGE	MEAN	MEDIAN	VARIANCE
0.7897703	2.771902	1.982131	1.780235	1.782253	0.3336386

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN HYPERCUBE SAMPLE FOR RAW DATA PAGE 1

1	1.0000					
2	-0.0121	1.0000				
3	-0.0050	-0.0121	1.0000			
4	0.0042	0.0105	-0.0103	1.0000		
5	-0.0007	0.0074	-0.0030	-0.0016	1.0000	
6	-0.0180	0.0724	0.0070	-0.0582	-0.0394	1.0000

7	-0.1086	-0.1241	0.0291	-0.0309	-0.1030	-0.0032	1.0000					
8	0.0002	-0.0006	0.0086	0.0084	0.0029	0.0041	0.0204	1.0000				
9	0.0065	-0.0072	0.0075	0.0022	0.0071	-0.0541	-0.0758	0.0153	1.0000			
10	0.0035	0.0043	-0.0063	0.0062	-0.0101	0.0310	-0.0191	0.0108	0.0018	1.0000		
11	0.0468	-0.0164	-0.0226	0.0110	-0.0057	0.0827	0.0240	-0.0168	0.0142	-0.0431	1.0000	
12	0.0099	-0.0099	0.0160	-0.0309	-0.0194	0.0040	0.0148	-0.0093	-0.0153	0.0117	0.0086	1.0000
	1	2	3	4	5	6	7	8	9	10	11	12

VARIABLES

THE VARIANCE INFLATION FACTOR FOR THIS MATRIX IS 1.05

TITLE SDB Name = INFIL.SDB , Ver = 1 12/01/99 17:58:05

CORRELATIONS AMONG INPUT VARIABLES CREATED BY THE LATIN HYPERCUBE SAMPLE FOR RANK DATA PAGE 1

1	1.0000											
2	-0.0177	1.0000										
3	-0.0097	0.0432	1.0000									
4	0.0330	0.0041	-0.0076	1.0000								
5	-0.0011	-0.0263	0.0272	0.0109	1.0000							
6	0.0056	-0.0069	0.0359	0.0131	0.0276	1.0000						
7	-0.0394	0.0051	0.0212	-0.0045	0.0126	0.0008	1.0000					
8	-0.0325	0.0420	-0.0150	-0.0459	-0.0076	0.0227	-0.0610	1.0000				
9	-0.0032	-0.0268	-0.0115	0.0025	-0.0192	0.0083	0.0021	0.0513	1.0000			
10	0.0442	0.0529	0.0334	-0.0153	0.0084	-0.0095	0.0338	-0.0228	-0.0400	1.0000		
11	0.0246	-0.0147	-0.0378	-0.0008	-0.0045	0.0242	0.0031	-0.0322	0.0200	-0.0386	1.0000	
12	0.0117	0.0168	0.0022	-0.0239	-0.0349	0.0010	0.0088	0.0132	-0.0117	0.0038	0.0082	1.0000
	1	2	3	4	5	6	7	8	9	10	11	12

VARIABLES

THE VARIANCE INFLATION FACTOR FOR THIS MATRIX IS 1.01

ATTACHMENT XII

OUTPUT RESULTS FROM ANALOG RATE MAPS SUPPLIED BY USGS OVER MODELED RECTANGULAR REGION

This attachment contains the GROPECDB displays of the analog maps, spatially averaged over the modeled rectangular region, corresponding to the analog infiltration rate maps produced by the USGS for the low, middle, and high glacial transition climates. The $\log_{10}$ of these bolded values (Tables XII-1 – XII-3) are used in calculating the weighting factors as presented in Table 6-2.

TABLE XII-1. GROPECDB Display of output data showing result for lower bound glacial transition analog infiltration rate map

```
                                GROPECDB_PA96

 GGGGG  RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB         PPPPPP
GG  GGG RRRRR   OO   OO PP      EE      CC      DD   DD BB   BB         PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB         PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB  _______ PP

                        GROPECDB_PA96 Version 2.12
                         PROD PA96 Built 06/27/96
                         Sponsored by Amy Gilkey

                       Run on 03/04/00 at 10:42:17
                   Run on ALPHA AXP BEATLE OpenVMS V7.2-1

Database:
I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.GLACIAL.LOW]POSTINF_LOW.CDB;1
Written on:  11/30/99 14:02:49

CAMDAT Version:   1        (EXODUS Version:   1)

GLOBAL TIME STEP VARIABLES

   Step       Time        NCELMEAN       INF_MEAN       RUN_MEAN
      1   0.00000E+00   4.78500E+03   2.36104E+00   0.00000E+00
```

TABLE XII-2. GROPECDB Display of output data showing result for MIDDLE glacial transition analog infiltration rate map

```
                                GROPECDB_PA96

 GGGGG  RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB ------- PPPPPP
GG  GGG RRRRR   OO   OO PP      EE      CC      DD   DD BB   BB         PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB         PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB ------- PP

                        GROPECDB_PA96 Version 2.12
                         PROD PA96 Built 06/27/96
                         Sponsored by Amy Gilkey

                       Run on 03/04/00 at 09:08:14
                  Run on ALPHA AXP BEATLE OpenVMS V7.2-1

Database:
I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.GLACIAL.MEAN]POSTINF_R001.CDB
;1
Written on:  11/30/99 10:36:47

CAMDAT Version:   1        (EXODUS Version:   1)

PROPERTIES

Element Block 1)    "FOOTPRNT"     1=ID    75 elements (1..75)
      8-node     5 attributes    13 properties
   BRPOROS      BRZDEPTH     SOILDEPM     PRECIPM      POTETMUL     BRPERM
   SOILPERM     ETCOEFFA     ETCOEFFB     FLAREA       ATEMP1       SNOPAR1
   SUBPAR1
  2.95100E-02  3.74700E+00  1.23000E+00  1.14400E+00  8.52300E-01  1.36700E+01
  3.42700E+00  1.12100E+01  1.10200E+00  3.04400E-01  1.73000E+01  2.15800E+00
  1.57100E-01

Element Block 2)    "OUTSIDE "     2=ID    25 elements (76..100)
      8-node     5 attributes    13 properties
   BRPOROS      BRZDEPTH     SOILDEPM     PRECIPM      POTETMUL     BRPERM
   SOILPERM     ETCOEFFA     ETCOEFFB     FLAREA       ATEMP1       SNOPAR1
   SUBPAR1
  2.00000E-02  1.50000E+00  1.00000E+00  1.00000E+00  1.00000E+00  1.00000E+00
  1.00000E+00 -1.50000E+00  1.26000E+00  5.00000E-01  1.73000E+01  1.78000E+00
  1.00000E-01

GLOBAL TIME STEP VARIABLES

   Step        Time        NCELMEAN       INF_MEAN       RUN_MEAN
      1   0.00000E+00   4.78500E+03   2.07892E+01   0.00000E+00
```

TABLE XII-3. GROPECDB Display of output data showing result for UPPer bound glacial transition analog infiltration rate map

```
                                GROPECDB_PA96

 GGGGG  RRRRRR   OOOOO  PPPPPP  EEEEEEE  CCCCC  DDDDDD  BBBBBB          PPPPPP
GG   GG RR   RR OO   OO PP   PP EE      CC   CC DD   DD BB   BB         PP   PP
GG      RR   RR OO   OO PP   PP EE      CC      DD   DD BB   BB         PP   PP
GG      RRRRRR  OO   OO PPPPPP  EEEEE   CC      DD   DD BBBBBB ------- PPPPPP
GG  GGG RRRRRR  OO   OO PP      EE      CC      DD   DD BB   BB         PP
GG   GG RR  RR  OO   OO PP      EE      CC   CC DD   DD BB   BB         PP
 GGGGG  RR   RR  OOOOO  PP      EEEEEEE  CCCCC  DDDDDD  BBBBBB -------- PP

                        GROPECDB_PA96 Version 2.12
                         PROD PA96 Built 06/27/96
                         Sponsored by Amy Gilkey

                       Run on 03/04/00 at 10:25:29
                   Run on ALPHA AXP BEATLE OpenVMS V7.2-1

Database:
I1:[000000.RDMCCUR.INEEL_PA98.YMP_99.INFILTRATION.POSTINFIL.GLACIAL.HIGH]POSTINF_HIGH.CDB
;1
Written on:  12/01/99 09:37:14

CAMDAT Version:   1        (EXODUS Version:   1)

GLOBAL TIME STEP VARIABLES

   Step        Time        NCELMEAN       INF_MEAN       RUN_MEAN
      1   0.00000E+00   4.78500E+03   3.92174E+01   0.00000E+00
```