1/58

SOFTWARE RELEASE NOTICE

1. SRN Number: PA-SRN-180

2. Project Title: TPA Postprocessor Version 3.2 β

Project No. 20-1402-762

3. SRN Title: TPA Postprocessor Version 3.2 β

4. Originator/Requestor: Bruce Mabrito

Date: 11/25/98

5. Summary of Actions

■ Release of new software

☐ Release of modified software:

  ☐ Enhancements made

  ☐ Corrections made

☐ Change of access software

☑ Software Retirement GW 2/12/2003

6. Persons Authorized Access

Name	Read Only/Read-Write	Addition/Change/Delete
Hollis A. Thomas	RW	Addition
Sitakanta Mohanty	RW	Addition
Tim McCartin (NRC)	RW	Addition
M. Rose Byrne (NRC)	RW	Addition

7. Element Manager Approval: Sitakanta Mohanty for G. Wittmeyer Date: 11/25/98

8. Remarks:

CNWRA Form TOP-6 (05/98)

SOFTWARE SUMMARY FORM

01. Summary Date: 11/25/98

02. Summary prepared by (Name and phone):
Hollis A. Thomas (210) 522-4958

03. Summary Action:
Release of
New Software

04. Software Date: 11/18/98

05. Short Title: TPA Postprocessor Version 3.2 β

06. Software Title: TPA Postprocessor Version 3.2 β

07. Internal Software ID:
None

08. Software Type:

☐ Automated Data System

☒ Computer Program

☐ Subroutine/Module

09. Processing Mode:

☒ Interactive

☐ Batch

☐ Combination

10. Application Area:

a. General:
☐ Scientific/Engineering ☒ Auxiliary Analyses
☐ Total System PA
☐ Subsystem PA ☐ Other

b. Specific:

11. Submitting Organization and Address:

CNWRA/SwRI
6220 Culebra Road
San Antonio, TX 78228

12. Technical Contact(s) and Phone:

Hollis A. Thomas (210) 522-4958
Sitakanta Mohanty (210) 522-5185

13. Software Application: Plot data output form the TPA code.

14. Computer Platform:
SUN Workstation
Windows PC

15. Computer Operating System: UNIX or Windows 95/NT

16. Programming Language(s): JAVA

17. Number of Source Program Statements: Approx. 3500 lines

18. Computer Memory Requirements: 30 Mb HD
32 Mb RAM

19. Tape Drives: N/A

20. Disk Units: N/A

21. Graphics: N/A

22. Other Operational Requirements: Requires installation of JAVA Development Kit.

23. Software Availability:
☐ Available ☒ Limited ☐ In-House ONLY

24. Documentation Availability:
☒ Available ☐ Preliminary ☐ In-House ONLY

25.
Software Developer: Hollis A. Thomas Date: 11/25/98

CNWRA Form TOP-4-1 (05/98)

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES

DESIGN VERIFICATION REPORT FOR CNWRA SOFTWARE:
TPA Version 3.2 Post-Processor (PP) Beta

November 25, 1998

Total-System Performance Assessment (Scientific and Engineering Software) Version 3.2 Post-Processor (PP) Beta

NOTE: This version of the TPA Software contains the integration of TPA Version 3.2 with JAVA to provide displays and increase capabilities in that area. An electronic scientific notebook assigned to Sitakanta Mohanty has been utilized as the change documentation method.

1. **This Design Verification Report is prepared by:** Bruce Mabrito in conjunction with Hollis Thomas.
Full Title of CNWRA scientific and engineering software: Total-System Performance Assessment (TPA) Version 3.2 Post-Processor (PP) Beta.
Demonstration work station: PC Pentium II Processor in conjunction with the MAMMOTH server from S. Mohanty's office.
Operating System: Windows NT 4.0

2. **Software Requirements Description and any changes thereto approved by Element Manager?**
[✓ YES] NO N/A

NOTE: A very straightforward and short SRD was prepared by H. Thomas (of SwRI Division 10) and was approved after-the-fact by the CNWRA PA Element Manager.

3. **Software Development Plan (SDP) and any changes have been approved by the Element Manager?**
[✓ YES] NO N/A

NOTE: A very straightforward and short SDP was prepared by H. Thomas and was approved after-the-fact by the CNWRA PA Element Manager.

4. **Design and Development**
Module-level testing is documented in either scientific notebooks or in Software Change Reports?
[✓ YES] NO N/A

NOTE: Note: An electronic scientific notebook (No. 170) was utilized and contains module level documentation.

5. Is the CNWRA scientific and engineering software developed in accordance with the conventions described in the SDP?

[✓ YES] NO N/A

6. Is the CNWRA software documented internally?

[✓ YES] NO N/A

Does the primary program header contain the following information:

A. Program title, Developed for (Customer), Office/Division/Date/Customer Contact/Telephone number, Software Developer, Telephone number, titles of Associated Documentation/Designator, and the Disclaimer Notice?

[✓ YES] NO N/A

B. Source code module header information provides Program Name, Client Name, Contract Reference, Revision number?

[✓ YES] NO N/A

NOTE: The latest CNWRA/SwRI Contract No. (NRC-02-97-009) was not reflected in the source code module header of TPA Version 3.2 PP Beta. The software developer was made aware of this but it was agreed not to change it at this late date in the development of the code. Other requirements were fulfilled.

7. Software designed so that individual runs are uniquely identified by Date, Time, Name of software and version? [✓ YES] NO N/A

8. The physical labeling on the software or the referenced list has Program Name/Title, Module/Name/Title, Module Revision, File Type (i.e. ASCII, OBJ, EXE), Recording Date and Operating System of the Supporting Hardware?

[✓ YES] NO N/A

9. Users' Manual

Is there a Users' Manual for the software?

[✓ YES] NO N/A

NOTE: The TPA SwRI Div. 10 Version 3.2 PP Beta Users' Manual (dated Nov. 20, 1998) was available during the Design Verification activities. A separate CNWRA TPA V 3.2 PP Beta handout will be written by the CNWRA and sent to the NRC later.

Are there basic instructions for the use of the software?

[✓ YES] NO N/A

10. Acceptance Testing

Does the acceptance testing demonstrate whether or not requirements in the SDP have been fulfilled?

[✓ YES] NO N/A

NOTE: TPA V3.2 PP Beta was compiled, linked, executed and tested on Kender. Since the CNWRA has no direct access to CRADAL (the future NRC server), that testing could not be conducted from the San Antonio location.

Has acceptance testing been conducted for each intended computer platform and operating system?

[✓ YES] NO N/A

NOTE: Acceptance testing on Sun platforms with the Solaris O.S. was performed Summaries are in the electronic scientific notebook pages.

Have installation tests been performed on the target platform?

[✓ YES] NO N/A

11. Configuration Control

Is the Software Summary Form completed and signed?

[✓ YES] NO N/A

If no, explain:

12. Is a software technical description prepared, documenting the essential mathematical and numerical basis?

[✓ YES] NO N/A

If no, explain: The technical description is given in the Users' Manual.

13. Is the source code available (or, is the executable code available in the case of commercial codes)?

[✓ YES] NO N/A

NOTE: For the TPA V 3.2 PP Beta, the answer is yes.

14. Have all the script/make files and executable files been submitted to the Software Custodian?

YES NO [✓ N/A]

Hollis A. Thomas 11/25/98

Hollis Thomas **Date**

CNWRA TPA Software Co-Developer

11/25/98

Bruce Mabrito **Date**

CNWRA Software Custodian

Attachments/
Original to: Software Folder
cc: CNWRA Software Developer/Cognizant EM/S. Mohanty

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
QUALITY ASSURANCE
SURVEILLANCE REPORT

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

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

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

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

QA REPRESENTATIVE: B. Mabrito

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

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

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

UNSATISFACTORY FINDINGS: None.

NONCONFORMANCE REPORT NO.: None.

ATTACHMENTS: Attachment A.

RECOMMENDATIONS/ACTIONS: N/A.

APPROVED:
CENTER DIRECTOR OF QUALITY ASSURANCE

DATE: 6/12/2000

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

CNWRA FORM QAP-8 (4/93)

```
/*
Program Name:          Total-System Performance Assessment Plotting Tool
File Name:             Plotter.java
File Date:             11/18/98
Release Version:       1.0beta

Client Name:           USNRC
                       U. S. Nuclear Regulatory Commission
                       NRC Office of Nuclear Material Safety and
Safeguards
                       Division of Waste Management
Contract Number:       NRC 02-93-005

NRC Contact            Tim McCartin (301) 415-6681

CNWRA Contact:         Sitakanta Mohanty (210) 522-5185
                       Center for Nuclear Waste Regulatory Analyses
                       San Antonio, Texas 78238-5166
                       smohanty@swri.edu

Documentation:         "Total-System Performance Assessment (TPA) Plotting
Tool
                       User's Guide", Center for Nuclear Waste Regulatory
                       Analyses (in preparation)

NUREG-Series Designator: N/A

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

                       D I S C L A I M E R

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

"This computer code/material was prepared as an account of work
performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA)
for the Division of Waste Management of the Nuclear Regulatory
Commission (NRC), an independent agency of the United States
Government.  Neither the developer(s) of the code nor any of their
sponsors make any warranty, expressed or implied, or assume any legal
liability or responsibility for the accuracy, completeness, or
usefulness of any information, apparatus, product or process
disclosed, or represent that its use would not infringe on privately-
owned rights."

 "In no event unless required by applicable law will the sponsors
or those who have written or modified this code, be liable for
damages, including any lost profits, lost monies, or other special,
incidental or consequential damages arising out of the use or
inability to use the program (including but not limited to loss of
data or data being rendered inaccurate or losses sustained by third
parties or a failure of the program to operate with other programs),
even if you have been advised of the possibility of such damages or
for any claim by any other party."

= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
*/


import java.io.*;
import java.awt.*;
import java.awt.event.*;
```

SOFTWARE USER GUIDE – TPA Version 3.2PPβ (Post –Processor)

Introduction

The TPA Plotting Tool is a program designed to produce a variety of standard graphs from data generated by the Total-system Performance Analysis (TPA) program. It is written in Java and will run on either a Unix or MS Windows platform. The program consists of an intuitive, easy to use interface which allows a user to select a plot to display.

System Requirements

Software requirements: Windows 95, Windows 98, or Windows NT (4.0), or Unix operating system; Java Development Kit (JDK 1.1.7 or greater); TPA Plotting Tool classes; JClass Chart classes. The JDK is available free on the Internet at `www.java.sun.com`, and the last two items are included in the software delivery. Data files generated by the TPA code are also required, although sample data files are included with the delivery.

Hardware requirements: Pentium processor (166 MHz or better) or equivalent Unix machine, 32MB RAM, 30MB hard disk space.

Types of Plots Available:

1. The maximum time plot of "Average WP Temperature vs. Time" is a time history of the WP temperature values from the NFENV module that are reported in the nearfld.res file for every 10 TPA time steps. The WP temperature is averaged over all subareas with equal weighting for each subarea. The compliance time plot of "Average WP Temperature vs. Time" uses the same values as the maximum time plot of "Average WP Temperature vs. Time". The TPA code does not write a nearfld_c.res file.

2. The maximum time plot of "Average RH vs. Time" is a time history of the relative humidity (RH) values from the NFENV module that are reported in the nearfld.res file for every 10 TPA time steps. The RH is averaged over all subareas with equal weighting for each subarea. The compliance time plot of "Average RH vs. Time" uses the same values as the maximum time plot of "Average RH vs. Time". The TPA code does not write a nearfld_c.res file.

3. The maximum time plot of "Average Cl Concentration vs. Time" is a time history of the Cl- concentration values from the NFENV module that are reported in the nearfld.res file for every 10 TPA time steps. The Cl- concentration is averaged over all subareas with equal weighting for each subarea. The compliance time plot of "Average Cl Concentration vs. Time" uses the same values as the maximum time plot of "Average Cl Concentration vs. Time". The TPA code does not write a nearfld_c.res file.

4. The maximum time plot of "Average Infiltration Rate" is a time history of three infiltration rates: the average infiltration rate from UZFLOW, the infiltration rate after reflux from the NFENV module, and the infiltration rate after diversion (using the Fmult and Fow parameters). These values are reported in the infilper.res file for every 10 TPA time steps. The infiltration rates are averaged over all subareas with equal weighting for each subarea. The compliance time plot of "Average Infiltration Rate" uses the same values as the maximum time plot of the "Average Infiltration Rate". The TPA code does not write an infilper_c.res file.

5. The maximum time plot of "Total Dose vs. Time for Realization 1" is the time history reported in totdose.res of the total dose to the receptor group from all groundwater and ground surface radionuclides for Realization 1. The total dose is the sum of the individual radionuclide doses calculated in DCAGW and DCAGS at each time step in the maximum time period. These values are reported in the totdose.res file and provide total dose over the maximum time at each TPA time step. The compliance time plot of "Total Dose vs. Time for Realization 1" is the time history reported in totdose.res of the total dose to the receptor group from all groundwater and ground surface radionuclides for Realization 1. The total dose is the sum of the individual radionuclide doses calculated in DCAGW and DCAGS at each time step in the compliance period. These values are reported in the totdose_c.res file and provide total dose over the compliance period at each TPA time step.

9/58

6. The maximum time plot of "EBS Peak Release Rate vs. Time (Tc-99, subarea 1)" is a scatter plot of the peak release rate of a radionuclide (Tc-99) from the engineered barrier system (EBS) and the corresponding time of the peak release rate for all realizations. These values are computed in EBSREL and are reported in the pkreltim.res file over the maximum time at each TPA time step. The compliance time plot of "EBS Peak Release Rate vs. Time (Tc-99, subarea 1)" is a scatter plot of the peak release rate of a radionuclide (Tc-99) from the engineered barrier system (EBS) and the corresponding time of the peak release rate for all realizations. These values are computed in EBSREL and are reported in the pkreltim_c.res file over the compliance period at each TPA time step.

7. The maximum time plot of "GW Peak Dose vs. Time of Peak for Tc-99" is a scatter plot of the peak groundwater dose of a radionuclide (Tc-99) to a receptor group and the corresponding time of the peak groundwater dose for all realizations. These values are computed in DCAGW and are reported in the npkdoset.res file over the maximum time at each TPA time step. The compliance time plot of "GW Peak Dose vs. Time of Peak for Tc-99" is a scatter plot of the peak groundwater dose of a radionuclide (Tc-99) to a receptor group and the corresponding time of the peak groundwater dose for all realizations. These values are computed in DCAGW and are reported in the npkdoset.res file over the compliance period each TPA time step.

8. The maximum time plot of "GW Peak Total Dose vs. Time" is a scatter plot of the peak total groundwater dose to a receptor group and the corresponding time of the peak total groundwater dose for all realizations. The peak total groundwater dose is the maximum of the total groundwater dose which is the sum of individual radionuclide doses calculated in DCAGW. The peak total groundwater dose and the corresponding time of the peak dose over the maximum time are reported in the gwpkdos.res file. The compliance time plot of "GW Peak Total Dose vs. Time" is a scatter plot of the peak total groundwater dose to a receptor group and the corresponding time of the peak total groundwater dose for all realizations. The peak total groundwater dose is the maximum of the total groundwater dose which is the sum of individual radionuclide doses calculated in DCAGW. The peak total groundwater dose and the corresponding time of the peak dose over the compliance period are reported in the gwpkdos_c.res file.

9. The maximum time plot of "CCDF of Air Peak Total Dose" is a CCDF of the peak total ground surface dose to a receptor group from an extrusive volcanic event. The peak total ground surface dose is the maximum of the total ground surface dose which is the sum of individual radionuclide doses calculated in DCAGS. The peak total ground surface dose over the maximum time is reported in the airpkdos.res file. The compliance time plot of "CCDF of Air Peak Total Dose" is a CCDF of the peak total ground surface dose to a receptor group from an extrusive volcanic event. The peak total ground surface dose is the maximum of the total ground surface dose which is the sum of individual radionuclide doses calculated in DCAGS. The peak total ground surface dose over the compliance period is reported in the airpkdos.res file.

10. The "Histogram of Average WP Failure Time" plot presents a histogram of the average WP failure time from corrosion and disruptive events (faulting, intrusive volcanic and seismic events). The WP failure times for all realizations is placed into bins to generate data used to construct the histogram. The frequency in the histogram plot represents only the fraction of realizations for which the average WP failure time (not the number of failed waste packages) is within a specified time interval. The average WP failure time for a realization is computed by weighting the time of corrosion and disruptive events reported in the wpsfail.res file with the corresponding number of failed WPs. The compliance time plot of "Histogram of Average WP Failure Time" uses the same values as the maximum time plot of the "Histogram of WP Failure Time". The TPA code does not write an wpsfail_c file.

11. The maximum time plot of "CCDF of Total EPA Normalized Release" is a CCDF of the sum of the groundwater and ground surface EPA normalized releases for all realizations. The EPA normalized release is computed using SZFT results of the total amount of a radionuclide released from the saturated zone over the maximum time for the groundwater and VOLCANO results of the total amount of a radionuclide released from the extrusive volcanic event for the ground surface. These releases are normalized using the EPA release limit of the radionuclide. The total EPA normalized release for a realization is computed by summing the groundwater and ground surface releases for each radionuclide over all radionuclides. The total EPA normalized release over the maximum time is reported in the relccdf.res file. The compliance time plot of "CCDF of Total EPA Normalized Release" is a CCDF of the sum of the groundwater and ground surface EPA normalized releases for all realizations. The EPA normalized-release is computed using SZFT results of the total amount of a radionuclide released from the

saturated zone over the compliance period for the groundwater and VOLCANO results of the total amount of a radionuclide released from the extrusive volcanic event for the ground surface. These releases are normalized using the EPA release limit of the radionuclide. The total EPA normalized release for a realization is computed by summing the groundwater and ground surface releases for each radionuclide over all radionuclides. The total EPA normalized release over the compliance period is reported in the relccdf.res file.

12. The maximum time plot of "CCDF of GW EPA Normalized Release" is a CCDF of the groundwater EPA normalized releases for all realizations. The groundwater EPA normalized release is computed using SZFT results of the total amount of a radionuclide released from the saturated zone over the maximum time. The release is normalized using the EPA release limit of the radionuclide. The total EPA normalized release in the realization is computed by summing the groundwater releases for each radionuclide over all radionuclides. The total EPA normalized release over the maximum time is reported in the gwccdf.res file. The compliance time plot of "CCDF of GW EPA Normalized Release" is a CCDF of the groundwater EPA normalized releases for all realizations. The groundwater EPA normalized release is computed using SZFT results of the total amount of a radionuclide released from the saturated zone over the compliance period. The release is normalized using the EPA release limit of the radionuclide. The total EPA normalized release in the realization is computed by summing the groundwater releases for each radionuclide over all radionuclides. The total EPA normalized release over the compliance period is reported in the gwccdf.res file.

13. The maximum time plot of "CCDF of EBS, UZ, SZ Releases for Tc-99" consists of three CCDFs of the total release from the engineered barrier system (EBS), the saturated zone (SZ), and the unsaturated zone (UZ) over the maximum time for one radionuclide (Tc-99). The three CCDFs are constructed from the total EBS,SZ, and UZ releases of the radionuclide for all realizations that are computed from EBSREL, SZFT, and UZFT results, respectively. The EBS, SZ, and UZ releases during the maximum time are reported in the cumrel.res file. The compliance time plot of "CCDF of EBS, UZ, SZ Releases for Tc-99" consists of three CCDFs of the total release from the engineered barrier system (EBS), the saturated zone (SZ), and the unsaturated zone (UZ) over the compliance period for one radionuclide (Tc-99). The three CCDFs are constructed from the total EBS,SZ, and UZ releases of the radionuclide for all realizations that are computed from EBSREL, SZFT, and UZFT results, respectively. The EBS, SZ, and UZ releases during the compliance period are reported in the cumrel_c.res file.

14. The maximum time plot of "CCDF of Average GWTT (UZ + SZ)" is a CCDF of the sum of the average groundwater travel times (GWTT) for the unsaturated zone (UZ) and saturated zone (SZ). These travel times represent the average GWTT for all subareas with equal weighting for each subarea. In each realization, the UZ GWTT is computed in UZFT and the SZ GWTT is calculated in SZFT for each subarea. The subarea averaged UZ and SZ GWTT values are reported in the gwttuzsz.res file. The compliance time plot of "CCDF of Average GWTT (UZ + SZ)" uses the same values as the maximum time plot of "CCDF of Average GWTT (UZ + SZ)". The TPA code does not write a gwttuzsz_c.res file.

15. The plot of "Expected Dose vs. Time" is a plot of the expected total dose as a function of time. The expected dose is the average of the total dose from all realizations at each time step. The expected total dose curve is not weighted by the scenario probability. The scenarios are specified in the tpa.inp file and are designated as oso, fso, and osv (i.e., oso, fso, and osv correspond to the basecase with seismicity, to faulting events, and to volcanic events, respectively). The time history of the expected doses is reported in the rgwsa.tpa file. The compliance time plot of "Expected Dose vs. Time" uses the same values as the maximum time plot of "Expected Dose vs. Time" during the compliance period. The TPA code does not write a rgwsa_c.tpa file.

Installation Notes

For Windows NT:

1. Copy the Plotter folder to c:\Plotter
2. Install JDK version 1.1.7 to c:\jdk1.1.7 by following the installation instructions. Get JDK at `www.java.sun.com`.
3. Copy jcchart300.jar to c:\jdk1.1.7\lib
4. Go to Control Panel ... System ... Environment
5. Append the following to the PATH variable (note the period at the end):

`c:\jdk1.1.7\bin;.`

6. Append the following to the CLASSPATH variable (note the period at the end) :
   `c:\jdk1.1.7\lib\classes.zip; c:\jdk1.1.7\lib\jcchart300.jar;.`
7. Restart Windows for the changes to take effect.
8. To run the program, get a DOS prompt and type:

```
cd c:\Plotter
javac Plotter.java
java Plotter
```

For Windows 95 and Windows 98:

1. Copy the Plotter folder to c:\Plotter
2. Install JDK version 1.1.7 to c:\jdk1.1.7 by following the installation instructions. Get JDK at `www.java.sun.com`.
3. Copy jcchart300.jar to c:\jdk1.1.7\lib
4. Add the following lines to c:\autoexec.bat

```
PATH c:\jdk1.1.7\bin;.
set CLASSPATH=c:\jdk1.1.7\lib\classes.zip;c:\jdk1.1.7\lib\jcchart300.jar;.
```

   [Note: if a PATH statement already exists, append `c:\jdk1.1.7\bin;.` to the end]
5. Restart Windows for the changes to take effect.
6. To run the program, get a DOS prompt and type:

```
cd c:\Plotter
javac Plotter.java
java Plotter
```

For Unix:

1. Copy the Plotter folder to /home/joeuser/Plotter (for example)
2. Install JDK version 1.1.7 to by following the installation instructions. Get JDK at `www.java.sun.com`
3. Copy jcchart300.jar to /home/joeuser/Plotter
4. Add the following line to the user's `.login` file (note the period at the end):

```
setenv CLASSPATH /bin:/home/joeuser/Plotter/jcchart.jar:.
```

5. Logout and login for the changes to take effect.
6. To run the program type:

```
cd /home/joeuser/Plotter
javac Plotter.java
java Plotter
```

User Support

For technical assistance, users may contact

Hollis A. Thomas
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510
(210) 522-4958
hthomas@swri.org

Sitakanta Mohanty
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510
(210) 522-5185
smohanty@swri.org

12/58

SOFTWARE REQUIREMENTS DESCRIPTION – TPA Version 3.2PPβ (Post –Processor)

1.0 SOFTWARE FUNCTION

The TPA Postprocessor application is a tool which will allow users to plot and view the outputs of a TPA run in graphical form. The graphical interface will present the user with a choice of approximately 15 views to plot, read the appropriate data file based on the user's selection, and pass the data to JClass Chart (a commercial Java class library) for plotting. Users will be able to choose to plot either maximum or compliance time data and will be able to print the graph if desired. The system will be able to generate the following plots:

- Average WP Temperature vs. Time
- Average RH vs. Time
- Average Cl Concentration vs. Time
- Average Infiltration Rate
- Total Dose vs. Time (Realization 1)
- EBS Peak Release Rate vs. Time (Tc-99, subarea 1)
- GW Peak Dose vs. Time of Peak (Tc-99)
- GW Peak Total Dose vs. Time
- CCDF of Air Peak Total Dose
- Histogram of Average WP Failure Time
- CCDF of Total EPA Normalized Release
- CCDF of GW EPA Normalized Release
- CCDF of EBS, UZ, and SZ Releases (Tc-99)
- CCDF of Average GWTT (UZ + SZ)
- Expected Dose vs. Time

2.0 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

The TPA code produces a series of data files (denoted with a `.res` or `.plt` extension) which the TPA postprocessor software reads and plots. The user's plot selection in the GUI determines which data file use. The core of the graphing engine is a set of commercial Java libraries called JClass Chart.

3.0 COMPUTATIONAL APPROACH

3.1 Data Flow and User Interface

The graphical user interface provides the following items:

- Menu: User can print a graph, get help, check the software version, or quit
- Time selection buttons: user can choose to display either a maximum or compliance time plot
- Plot selection buttons: user can select which graph to display
- Data path field: user can specify where the TPA-generated data files are located
- Plot area: the area where the graph is displayed

During program execution, typical data flow is as follows:

When the user chooses a plot, the program will read the data file corresponding to the selected *plot selection button* and *time selection button* from the directory specified in the *data path field* and will graph the data in the *plot area*. The user can print the graph using the command in the *menu.*

3.2 Hardware and Software Requirements

Target Platform: Windows-based PC or Unix-based workstation

Operating System: Windows 95/98, Windows NT, or Unix

Programming Language: Java

Software requirements: Windows 95, Windows 98, or Windows NT (4.0), or Unix operating system; Java Development Kit (JDK 1.1.7 or greater); TPA Plotting Tool classes; JClass Chart classes. The JDK is available free on the Internet at `www.java.sun.com`, and the last two items are included in the software delivery. Data files generated by the TPA code are also required, although sample data files are included with the delivery.

Hardware requirements: Pentium processor (166 MHz or better) or equivalent Unix machine, 32MB RAM, 30MB hard disk space.

3.3 Graphics Requirements

The software produces color output and requires a minimum resolution of 800x600.

3.4 Pre- and Post-Processors

The TPA Postprocessor plotting software requires data files generated by the TPA code.

4.0 REFERENCES

None

5.0 APPENDICES

None

Sitakanta Mohanty for G. Wittmeyer 11/25/98

APPROVED

Gordon Wittmeyer

SOFTWARE DEVELOPMENT PLAN – TPA Version 3.2PPβ (Post –Processor)

1.0 SCOPE

The TPA Postprocessor application is a tool which will allow users to plot and view the outputs of a TPA run in graphical form. The graphical interface will present the user with a choice of approximately 15 views to plot, read the appropriate data file based on the user's selection, and pass the data to JClass Chart (a commercial Java class library) for plotting. The application will be written in Java for cross platform portability and will operate on both Windows-based PCs and Unix-based workstations. Users will be able to print the graph if desired. A brief User Guide will accompany the software.

2.0 BASELINE ITEMS

- Graphical user interface code: generates the portion of the program visible to the user
- Plot generation code: displays the appropriate plot based on user input
- Code to read the data files: reads and formats data from a TPA run
- Test Data: sample TPA run data used to test the plotting program
- On-line Help Files: help the user can access from within the program
- User Guide: describes available plots, system requirements, and installation procedures

3.0 PROJECT MANAGEMENT

3.1 Work Breakdown Structure

Task	Estimated Labor Hours
Generate preliminary system requirements	5
Develop preliminary graphical user interface	20
Test and refine graphical interface operation	10
Develop code to read TPA output files	40
Test and refine data file read capability	10
Develop code to generate the required plots	100
Test and refine plot generation	30
Develop print capability	10
Test and refine print capability	5
Test cross platform portability	5
Develop on-line user help	10
Produce user guide	10
Modify graphical interface based on customer feedback	10
Modify plot format based on customer feedback	20
Add additional plots based on customer feedback	50
Prepare project management administrative paperwork	40
Total	375

3.2 Projected Schedule

Work will begin in mid-June 1998 and continue through the end of October 1998.

3.3 Staffing

One SwRI staff member working half-time for the duration of the project
One student working full-time from mid-June 1998 to the end of August 1998

3.4 Risk Management

Limitations of Java: Although Java is the appropriate language in which to produce this software, it is still a relatively new language and is still evolving and maturing. Some user requirements (e.g., some printing requirements) may have to be modified so Java can support them or delayed until the language develops the required capabilities. A new release of the language is expected before the end of 1998.

Limitations of JClass Chart. JClass Chart is the commercially-purchased product used to generate the graphs of the TPA data. Although the product is stable, well-written, and highly-rated, it may not have all the charting features the customer desires. Due to the time and funding limitations of this project, development will be dependant on the capabilities built into the class libraries of this product. Some desires/requirements of the user may have to modified slightly.

Staffing Limitations. No experienced Java developers are available to work on this project for the time required to complete it. An experienced Java programmer will act as a mentor to the system developer.

4.0 DEVELOPMENT PROCEDURES

4.1 Hardware and Software Resources

The software will be developed on a Pentium-based PC running Windows NT. It will be ported to and tested on a Pentium-based PC running Windows 95/98 and a Sun Sparc workstation running Solaris. These hardware resources are already available in-house and do not need to be purchased.

The system will require JClass Chart for graphing the TPA data. SwRI Division 10 has purchased this software on overhead at no cost to the customer.

The host machines require Sun's Java Development Kit (JDK) to run. This software is available free from Sun.

4.2 Software Development Lifecycle

Analysis: Determine input data format, formulate requirements for interface, determine plot output format
Design: Design layout of interface and plots
Product Development: Develop interface, data reading code, plot code, help files, and user guide
Iteration Release: Developers release a version of the software to users
Testing and User Feedback: Users provide developers feedback on "look and feel" and functionality of product. Developer uses this information and "loops back" to analysis, design, development and release
Final Delivery: Developers give a final version of software to users

4.3 Coding

The TPA Postprocessor Plotter will be written in Java using Sun's Java coding style.

4.4 Acceptance Testing and Analysis

SwRI Division 20 personnel will perform preliminary acceptance testing on the plotting software since they are familiar with the TPA code's output data, the end user requirements for the plotting software, and the expected form of the output plots. Testers will document results of testing and the proposed changes to the plotting software in scientific notebooks. The developers will use this information to make revisions to the program. After preliminary acceptance testing, the end user (the Nuclear Regulatory Commission) will use the software and provide comments and change requests to SwRI so that the code can be further modified.

5.0 CONFIGURATION MANAGEMENT PLAN (CMP)

5.1 Tools

Due to the relatively small size of the program, no special configuration management tools are required for this project.

5.2 Configuration Identification

The following items will be placed under configuration control:

Graphical user interface code: generates the portion of the program visible to the user
Plot generation code: displays the appropriate plot based on user input
Code to read the data files: reads and formats data from a TPA run
Test Data: sample TPA run data used to test the plotting program
On-line Help Files: help the user can access from within the program
User Guide: describes available plots, system requirements, and installation procedures

In order to place a particular release version under configuration control, the developers will create a folder named Plotter*mmdd* where *mmdd* is the date the folder was created. This folder will be archived.

5.3 Configuration Procedures

Due to the small size of both the software program and the development staff, there are no check-in/check-out procedures for this project.

Release versions of the software will be cleared through and approved by SwRI Division 20 personnel.

No official documentation such as an SCR is required for changes to the software during preliminary acceptance testing (*i.e.* the testing which SwRI Division 20 personnel perform), but these change requests will be recorded in the scientific notebooks of the tester and/or the developer.

Once the code is baselined and ready for delivery to the end user, the end user may request changes to the software using an SCR.

6.0 REFERENCES

None

7.0 APPENDICES

None

Sitakanta Mohanty for G. Wittmeyer

11/25/98

APPROVED
Gordon Wittmeyer

SOFTWARE REQUIREMENTS DESCRIPTION

17/58

SOFTWARE REQUIREMENTS DESCRIPTION - SUPERMODSIGN POST-PROCESSOR FOR TPA VERSION 3.2

By

Kevin Poor

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

Reviewed by:

James R. Weldy 4/8/99
James Weldy

Approved by:

Gordon Wittmeyer 4/8/99
Gordon Wittmeyer, Element Manager, Performance Assessment

SOFTWARE REQUIREMENTS DESCRIPTION - SUPERMODSIGN POST-PROCESSOR FOR TPA VERSION 3.2

1 SOFTWARE FUNCTION

Supermodsign is a post-processor tool for developing parameter trees from Total-system Performance Assessment (TPA) Version 3.2 code results. The Supermodsign post-processor development has an objective of making TPA code results more transparent and understandable than could be done with techniques already in use (e.g., plotting complementary cumulative distribution functions of code output, histograms, etc.). Parameter trees, as described in Jarzemba and Sagar (1999), are similar to event trees and are one technique that can make simulation results more transparent. Since event trees are commonly used to present risk assessment results for other complicated systems (e.g., nuclear reactors), and parameter trees are similar to event trees, it is expected that their application to repository performance assessment (PA) would also help clarify the results of complex PA models.

2 TECHNICAL BASIS: PHYSICAL AND MATHEMATICAL MODEL

The objective of the Supermodsign technique is to analyze the input vectors and the corresponding output vectors (that is, post-process the results) to estimate the relative sensitivity of the output to input parameters (taken singly and as a group) and thereby rank them. Relative sensitivities are estimated by developing a tree structure (which looks similar to an event tree but is not associated with a initiating event), with each limb of the tree representing a particular combination of parameters or a combination of system components. For convenience and to distinguish it from an event tree, we call it a parameter tree.

The approach to organizing TPA Version 3.2 code results into a tree-like structure is to group realizations into bins based on a commonality (in terms of their magnitudes) of their input parameters and output variable (e.g., peak dose in the compliance period of 10 kyr). Parameter values are treated as either a "+" or a "−" (a sign test) based on whether the value for a given realization of the parameter is greater or less than its median value for all realizations. Other branching criteria (e.g., mean, 90th percentile) are also possible. By grouping realizations in this manner, it is possible to determine which combination of parameters produces high or low doses. We are also able to define measures for sensitivities of individual parameters and for parameter groups. The parameter tree approach can be adapted to identify the initial set of most sensitive parameters without relying on traditional sensitivity analyses. This requires the implementation of the tree approach in a stepwise manner based on the value of a sensitivity factor for realizations on a particular branch of the tree. A more complete description of the assumptions and computational approach implemented in Supermodsign is contained in Jarzemba and Sagar (1999); see also appendix A.

3 COMPUTATIONAL APPROACH

3.1 DATA FLOW AND USER INTERFACE

The code will be designed to accept input data for parameter values from TPA files and Supermodsign input files. Supermodsign will write the results of the calculations to an output file. Input data to the Supermodsign code consist of sorting parameters identified in Supermodsign.in and modsign.in. Source information to be manipulated by Supermodsign is provided as output (either intermediate or final) from TPA version 3.2. Output data from Supermodsign will include a listing of those TPA input parameters analyzed (whether selected by the user or by the Supermodsign code), the number of realizations of TPA output that were above the overall parameter tree sign criterion value, the mean value of the output variable for any given bin, the percentage of the population mean of the output variable caused by realizations in a given bin, and an "importance factor," which is determined as the ratio of the contribution to the overall mean from realizations in that bin to the average contribution of the same number of realizations to the overall mean.

3.2 HARDWARE AND SOFTWARE REQUIREMENTS

Target Platform: Since the code will be written in standard FORTRAN90, there are no platform or system requirements

Operating System: See above

Programing Language: FORTRAN90

Software Requirements: None

Hardware Requirements: Any system capable of supporting the FORTRAN90 computer language.

3.3 GRAPHICS REQUIREMENTS

Supermodsign does not require graphics support and does not produce graphical outputs.

3.4 PRE- AND POST-PROCESSORS

The Supermodsign post-processor requires data files generated by the TPA code.

4 REFERENCES

Jarzemba, M.S., and B. Sagar. 1999. A Feasibility Study for a TPA Version 3.2 Event-Tree Post Processor, San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

APPENDIX A

A PARAMETER TREE APPROACH TO ESTIMATING SYSTEM SENSITIVITIES TO PARAMETER SETS

21/58

A PARAMETER TREE APPROACH TO ESTIMATING SYSTEM SENSITIVITIES TO PARAMETER SETS

by

Mark S. Jarzemba and Budhi Sagar

The Center for Nuclear Waste Regulatory Analyses
Southwest Research Institute
6220 Culebra Road
San Antonio, TX 78238-5166, USA

ABSTRACT

A technique for determining relative system sensitivity to groups of parameters and system components is presented. It is assumed that an appropriate parametric model to simulate system behavior is available and that some of the important parameters are stochastic variables that are described through probability distribution functions (PDFs). It is further assumed that the system behavior is simulated using *Monte Carlo* techniques that produce a realization of the system output(s) for every realization of the input parameter vector. The objective of our technique is to analyze the input vectors and the corresponding output vectors (that is, post-process the results) to estimate the relative sensitivity of the output to input parameters (taken singly and as a group) and thereby rank them. Relative sensitivities are estimated by developing a tree structure (which looks similar to an event tree but is not associated with a initiating event), each limb of the tree representing a particular combination of parameters or a combination of system components. For convenience and to distinguish it from the event tree, we call it the parameter tree.

To construct the parameter tree, the samples of input parameter values are treated as either a "+" or a "–" based on whether or not the sampled parameter value is greater than or less than a specified branching criterion (e.g., mean, median, percentile of the population). Partitioning the first parameter into a "+" or a "–" bin creates the first level of the tree containing two branches. At the next level, realizations associated with each first-level branch are further portioned into two bins using the branching criteria on the second parameter and so on until the tree is fully populated. Relative sensitivities are then inferred from the number of samples associated with each branch of the tree.

The parameter tree approach is illustrated by applying it to preliminary simulations of the proposed high-level radioactive waste repository at Yucca Mountain, NV. Using a Total System Performance Assessment Code called TPA, 4,000 realizations are obtained and analyzed. In the examples presented, groups of five important parameters, one for each level of the tree, are used to identify branches of the tree and construct the bins (i.e., realizations where all five of the important input parameters are

"+" are the contents of one bin, realizations where the first four parameters are "+" and the fifth is "–" form another bin, and so on). In the first example, the five important parameters are selected by more traditional sensitivity analysis techniques. This example shows that relatively few branches of the tree dominate system performance. In another example, the same 4,000 realizations are used but the most important five parameter set is determined in a stepwise manner (using the parameter tree technique) and it is found that these five parameters do not match the five of the first example. This important result shows that sensitivities based on individual parameters (i.e., one parameter at a time) may differ from sensitivities estimated based on joint sets of parameters (i.e., two or more parameters at a time).

The technique is extended using intermediate code outputs to define the branches of the tree. The intermediate outputs represent the behavior of a part of the system or that of a component or a subsystem. The intermediate outputs used in this example are the total cumulative radionuclide release (TCR) from the engineered barriers, unsaturated zone, and saturated zone. The TCR is defined as the time-integrated release of all radionuclide activity, measured in Curies (Ci), from each of the subsystems during a defined period (10 kyr in our example). The technique is found to be successful in estimating the relative influence of each of these three subsystems on the overall system behavior.

1 INTRODUCTION

Sensitivity analysis is a general term used to describe any study that quantifies how a given system output variable is modified with changes in system input variables. Many techniques are described in the literature for performing sensitivity analyses where changes in the output variable are compared to changes in a single input variable, one at a time. Among these techniques are: (i) stepwise multivariate regression[1,2] , (ii) differential analysis[3], (iii) rank regression[4,5], (iv) Kolmogorov-Smirnov test[6], and (v) signs test[6]. A limitation of all of these techniques is that they are generally not suitable for examining output variable sensitivity to input parameters in groups (i.e., jointly to two or more parameters). For example, it may be the case that a system output variable does not show a large sensitivity to each of two input parameters, but does show a large sensitivity when both parameters take on extreme values.

The purpose of this paper is to describe a parameter tree technique for examining total system output relative sensitivity to groups of input parameters. These parameter trees look similar to event trees but are not event trees because no real initiating event is associated with them. In this technique, the *Monte Carlo* (or stratified sampling) method is used to examine the possible outcomes of a scenario class for a given system. Bins of realizations (sometimes called subsets of scenarios where each represents a possible system outcome for a scenario class) are examined where the bins are determined by a commonality of their input parameter states (e.g., all sampled input parameters above their median value). Example applications of the technique are presented using preliminary simulations of the proposed high-level radioactive waste repository at Yucca Mountain (YM), NV. The Total-system Performance Assessment code called TPA[7,8] is used for the simulations.

2 GENERAL APPROACH: DEVELOPMENT OF THE PARAMETER TREE

Consider a system whose output (Y) is a random variable. In the following, we follow the convention of representing random variables in upper case and their particular samples (or realizations) in lower case symbols. In general, Y is a function of random parameters X_i, deterministic parameters d_k, and model assumptions a_m. We assume that the behavior of the system is simulated by appropriately sampling the random parameters and then computing the system output or realizations of Y for each parameter vector. For the purposes of this paper, which is to outline a method for analyzing simulation output to identify important random parameters and develop understanding of their relationship to the output, it is assumed that the decisions about appropriate model assumptions and deterministic parameters have been made *apriori*. As a result, we do not consider the dependence of Y on d_k and a_m any further and focus on the dependence of Y on the X_is only. Thus, for the jth realization of Y,

$$y_j = f\left(x_{1,j},\ x_{2,j},\ldots,x_{I,j}\right) \tag{1}$$

where I is the total number of sampled parameters in the model.

We want to analyze the outputs y_j to determine the sensitivity and correlations of Y to subgroups of the input parameters X_n, n = 1, 2, ..., N, where N<I. This can be done by developing a tree structure as explained in the following paragraphs.

Our approach for examining system output sensitivity to combinations of input parameters is to construct a parameter tree. Similar in appearance to an event tree, the parameter tree partitions parameter space into bins (each bin forming a branch of the tree) based on a partitioning (or branching) criterion. The simplest form of a branching criterion is a classification based on parameter magnitude which treats sampled values as either a "+" or a "–" depending upon whether the sampled value is greater or less than the branching criteria value. The event tree analogy is appropriate if one considers a "+" as a parameter failure and a "–" as a parameter success, or vice-versa. Figure 1 depicts a general parameter tree. To explain Figure 1 using a system model, a number of output realizations are generated for a given scenario class (e.g., airplane crashes into an operating nuclear reactor). Next, the realizations are partitioned into two subsets determined by whether the first important parameter (how to choose the first important parameter will be discussed later) is greater than or less than a specified level (e.g., airplane crashes of craft more or less massive than a fully fueled and loaded Boeing model 727, or of craft more massive than the national median value for fully fueled and loaded aircraft, or of craft more massive than the national mean value for fully fueled and loaded aircraft, etc.). Realizations with a high value are all treated as "+" and low as a "–", regardless of their position within the subset. The procedure is repeated in each of these two subsets with the next important parameter to be considered (i.e., the second-level parameter, say, the thickness of the reactor containment system) and so on until each of the important parameters is considered. This procedure determines 2^M bins of realizations where M is the number of important parameters. Note that not every sampled parameter in the system model need be considered if a subset of the sampled parameters satisfactorily explains system behavior of interest. In terms of our previous example, if an aircraft more massive than a fully loaded Boeing model 727 crashes into a reactor with a pressure vessel less than six inches thick always produces a system failure (i.e., a reactor breach and release of radioactive material) in all realizations, then no more variables need be considered.

In the following, we develop a formal explanation of this method. Let $\hat{X}_i$ be the median value of X_i, $\hat{Y}$ be the median value of Y, and I be the total number of sampled parameters. In this development, we use median values for partitioning criteria, but any other statistical or physical branching criterion could also be used, as will be explained later through an example. The first step in the procedure is to partition all of the realizations into two bins:

$$x_{1^+} = \left[\forall\, realizations\ with\ x_{1,j} \geq \hat{X}_1\right] \tag{2a}$$

$$x_{1^-} = \left[\forall\, realizations\ with\ x_{1,j} < \hat{X}_1\right] \tag{2b}$$

Assume that the two bins contain N_{1^+} and N_{1^-} members, respectively, where $N_{1^+} + N_{1^-} = N$ is the total number of samples or realizations. Note that when the partitioning criterion is the median value, $N_{1^+} = N_{1^-} = N/2$ but that will not be true for other branching criteria.

Now consider the N_{1^+} realizations of Y that are produced by the x_{1^+} set. From these N_{1^+} realizations, we select those that meet the following criterion:

$$y_{1^+} = \left[\forall \, realizations \; with \; y_j \geq \hat{Y} \,|\, x_{1,j} \in x_{1^+} \right] \tag{3}$$

Let the number of realizations satisfying this criteria be L_{1^+}. It follows that:

$$p_{1^+} = P\left\{ Y \geq \hat{Y} \,|\, X_1 \geq \hat{X}_1 \right\} = \frac{L_{1^+}}{N_{1^+}} \tag{4}$$

The second branch of the tree is associated with the y_{1^-} bin containing L_{1^-} members, where:

$$y_{1^-} = \left[\forall \, realizations \; with \; y_j \geq \hat{Y} \,|\, x_{1,j} \in x_{1^-} \right] \tag{5}$$

In this case, similar to equation (4),

$$p_{1^-} = P\left\{ Y \geq \hat{Y} \middle| X_1 < \hat{X}_1 \right\} = \frac{L_{1^-}}{N_{1^-}} \tag{6}$$

Equal values of p_{1^+} and p_{1^-} would imply that whether X_1 takes values greater or smaller than its median does not determine the bin into which Y values fall, thus indicating a lack of correlation or lack of sensitivity of Y to X_1. Consequently, a measure of relative sensitivity of Y with respect to X_1 can be constructed as $|p_{1^+} - p_{1^-}|$. It is noted that the proposed measure provides only relative sensitivity since it does not provide a precise description of the change in Y for a given change in X_1, as a measure for absolute sensitivity would provide. However, the relative sensitivity measure is sufficient for ranking important parameters. In general, one can partition the $x_{1.j}$ (and subsequent parameter realizations) into more than two bins but such a generalization will lead to a complicated tree structure (i.e., with potentially large numbers of branches per level) and is not pursued further in this paper.

The branching strategy explained above is now implemented for the second, third, and subsequent parameters until most of the output is sufficiently explained. For the second parameter, proceed as follows. Partition the bin x_{1^+} containing N_{1^+} realizations into two bins:

$$x_{1^+2^+} = \left[\forall\ realizations\ with\ x_{1,j} \geq \hat{X}_1 \cap x_{2,j} \geq \hat{X}_2\right] \tag{7a}$$

and

$$x_{1^+2^-} = \left[\forall\ realizations\ with\ x_{1,j} \geq \hat{X}_1 \cap x_{2,j} < \hat{X}_2\right] \tag{7b}$$

Similarly, the x_{1^-} bin can also be partitioned into two bins:

$$x_{1^-2^+} = \left[\forall\ realizations\ with\ x_{1,j} < \hat{X}_1 \cap x_{2,j} \geq \hat{X}_2\right] \tag{7c}$$

$$x_{1^-2^-} = \left[\forall\ realizations\ with\ x_{1,j} < \hat{X}_1 \cap x_{2,j} < \hat{X}_2\right]. \tag{7d}$$

and

Let the number of members in each of the four bins be $N_{1^+2^+}$, $N_{1^+2^-}$, $N_{1^-2^+}$, and $N_{1^-2^-}$ respectively.

The output realizations associated with members of a bin are now scrutinized to count the number of realizations in which $Y \geq \hat{Y}$. Thus, the four output bins associated with the four branches of the tree at the second parameter level are:

$$y_{1^+2^+} = \left[y_j \geq \hat{Y} \middle| x_{1,j}, x_{2,j} \in x_{1^+2^+} \right] \tag{8a}$$

$$y_{1^+2^-} = \left[y_j \geq \hat{Y} \middle| x_{1,j}, x_{2,j} \in x_{1^+2^-} \right] \tag{8b}$$

$$y_{1^-2^+} = \left[y_j \geq \hat{Y} \middle| x_{1,j}, x_{2,j} \in x_{1^-2^+} \right] \tag{8c}$$

$$y_{1^-2^-} = \left[y_j \geq \hat{Y} \middle| x_{1,j}, x_{2,j} \in x_{1^-2^-} \right] \tag{8d}$$

Let the number of realizations associated with the four bins of equation (8) be $L_{1^+2^+}$, $L_{1^+2^-}$, $L_{1^-2^+}$, and $L_{1^-2^-}$ respectively. Then at the second level of the tree, we can make the following probability statements,

$$p_{1^+2^+} = P\left\{ Y \geq \hat{Y} \middle| x_{1,j} \geq \hat{X}_1 \cap x_{2,k} \geq \hat{X}_2 \right\} = \frac{L_{1^+2^+}}{N_{1^+2^+}} \tag{9a}$$

and with similar interpretations,

$$p_{1^+2^-} = \frac{L_{1^+2^-}}{N_{1^+2^-}} \tag{9b}$$

$$p_{1^-2^+} = \frac{L_{1^-2^+}}{N_{1^-2^+}} \tag{9c}$$

$$p_{1^-2^-} = \frac{L_{1^-2^-}}{N_{1^-2^-}}. \qquad (9d)$$

If $p_{1^+2^+} = p_{1^+2^-}$ then the second parameter, X_2, (given $X_1 \geq \hat{X}_1$) has no influence on Y. Thus, relative sensitivities of X_2 can be partially measured by $|p_{1^+2^+} - p_{1^+2^-}|$ and $|p_{1^-2^+} - p_{1^-2^-}|$ for the cases of $X_1 \geq \hat{X}_1$ and $X_1 < \hat{X}_1$ respectively. The total relative sensitivity of Y to X_2 can be determined from:

$$S_{X_2} = |p_{1^+2^+} - p_{1^+2^-}| P\{X_1 \geq \hat{X}_1\} + |p_{1^-2^+} - p_{1^-2^-}| P\{X_1 < \hat{X}_1\} \qquad (10)$$

Also, $p_{1^+2^+}$ equal to $p_{1^-2^-}$, implies that whether the first two parameters together had high (greater than their medians) or low (smaller than their medians) values, there is an equal chance of producing a Y lower or higher than its median value. We propose the quantity $\frac{|p_{1^+2^+} - p_{1^-2^-}|}{1 - |p_{1^+2^+} - p_{1^-2^-}|}$ as a measure of the relative sensitivity of Y jointly to X_1 and X_2. For this example, we have assumed that both X_1 and X_2 are positively correlated with Y (i.e., large values of X_1 and X_2 lead to large values of Y and vice-verse). In general, this is not a valid assumption and input parameters can be positively or negatively correlated with the output variable. Hence, we now change our nomenclature for the joint relative sensitivity such that the coefficient is now defined as $\frac{|p_H - p_L|}{1 - |p_H - p_L|}$, where p_H and p_L are the greatest and least values of p among the bins.

In this formulation, the numerator represents the "distance" of the output variable from "perfect" non-correlation with the input parameter set (i.e., if Y has no correlation with the input parameter set under study, then p is the same in all bins and the numerator is zero). Similarly, the denominator represents the distance of the output variable from perfect correlation with the input parameter (i.e., if Y shows perfect correlation with the input parameter set under study, p is unity in the highest bin and zero in the lowest bin and the denominator is zero). With this formulation, the joint relative sensitivity is on the range $[0,\infty]$. This formulation can be extended to any number of parameters as is evident from the examples given later.

Another measure of influence of a subset of parameters may be defined through the contribution that realizations in a bin make to a specific statistic of the output. For example, one can compute the expected value of Y for realizations associated with each branch of the tree and compare these means to the overall mean of Y. Of course, statistics other than the mean can be used or probability distributions can be developed for each branch and compared to the overall probability distribution of Y. If for example, the probability of Y exceeding a certain limiting value (perhaps specified by regulations) is of interest, one could find the value of such exceedance probability for each branch and estimate (in a relative sense) the contribution that each parameter set makes to such a probability. Formally then, if T is a statistic (e.g., mean, mode, median, exceedance probability) of interest, for the second level of the tree, the ratios of $T_{1^+2^+}$, $T_{1^+2^-}$, $T_{1^-2^+}$, $T_{1^-2^-}$ to T of Y as a whole provide measures of relative sensitivity.

Consider now the earlier suggestion that the branching criterion can be something other than the magnitude of a parameter. One of the more useful possibilities is to envision the system as being made up of several components such that the output from one component becomes an input to the second and so on as indicated in Figure 2. With this conceptualization, the branching criterion can be stated in terms of the magnitude of the output of a component. In this case, each branch of the tree will represent the contribution of a component or a set of components to overall system performance. Relative sensitivity measures can be defined in exactly the same manner as explained above.

In general, the list of important (or most important) parameters is not known *apriori*, to develop such a list is, in fact, an important aspect of sensitivity analysis. The rest of this paper presents examples of the method using the U.S. Nuclear Regulatory Commission (NRC) TPA code, which was developed to evaluate the proposed High-Level radioactive Waste (HLW) repository at Yucca Mountain, NV. In these examples, the list of important parameters is determined in two ways: (i) using traditional sensitivity analyses, and (ii) using the parameter tree approach in a stepwise manner.

3 EXAMPLES OF PARAMETER TREE APPLICATIONS

This section provides example applications of the parameter tree approach. These examples use simulation data developed using the NRC TPA code. Several trees are presented, each using different branching criteria for the important input parameters. A stepwise implementation of the approach is also presented.

3.1 BACKGROUND

The TPA code used in these examples was jointly developed by the NRC and the Center for Nuclear Waste Regulatory Analyses. The TPA code has been used to conduct more traditional sensitivity analyses of repository performance[7,8]. In summary, this code includes models to predict the degradation and disruption of emplaced waste packages (WPs) loaded with commercial spent nuclear fuel, transport of radionuclides in groundwater to locations down gradient, and subsequent radiation doses that may occur from contaminated groundwater over long time periods (e.g., 10 kyr). Figure 3 provides a simplified description of the repository system and of how PA is conducted. Since the current version of this code has many hundreds of parameters with 244 being sampled in the nominal case input data set[6], the results of the more traditional analyses are used as a starting point in the first example in order to limit the number of input parameters investigated.

For the purposes of this paper, the output variable of the model is the peak annual total effective dose equivalent (peak dose) in 10 kyr following repository closure. Input variables to the model are numerous[6]. Figure 4 shows the results of a stepwise multilinear regression[1,2] of the residual sum of squares versus the number of parameters included in a multilinear fit of the logarithm of peak dose versus the logarithm of input parameters where the input parameters (i.e., the X_is) also appear only in first order in the fit [e.g., as in equation(1)]. This figure was generated using the S-plus statistical software package[9]. Because the data range over orders of magnitude and the results of the model are largely multiplicative rather than additive in the input mechanisms modeled (i.e., the calculated peak doses could be thought of as the product of release from the engineered barrier system times protection afforded from the geosphere surrounding the WP, times a factor that converts releases from the geosphere to dose), fits to the logarithm of the variables tend to produce better results. The form of the fitting function is:

$$\log[\mathrm{Y}] = m_1 \log[\mathrm{X}_1] + m_2 \log[\mathrm{X}_2] + \cdots + m_I \log[\mathrm{X}_I] + b \qquad (11)$$

where the m_i are the slopes and b is the intercept. All other variables are as previously defined. The stepwise regression routine in S-plus organizes the results so that the parameter that explains the greatest reduction in residual sum of squares (in the multilinear fit) appears first. As can be seen from Figure 4, only about five sampled parameters provide significant reduction in the residual sum of squares, hence initially these are the important input parameters. A short description of these parameters is provided in the following paragraph.

The first-level parameter, I_0, is the areal average mean annual infiltration of groundwater at the start of the simulation period. Infiltration at later times is modeled as a linear multiple of the starting infiltration depending on climate change. The second and fourth parameters, F_{ow} and F_{mult}, account for flow diversion or funneling in the unsaturated zone (UZ) above the repository and in the near field environment toward or away from the WP. The third parameter, WP_{def}, is the fraction of initially defective

WPs. The fifth parameter, SA_{wp}, is the fraction of the subarea that experiences wetting from infiltrating groundwater. Since flow in the UZ is primarily in fractures, this fraction may be less than unity. A multilinear regression also showed that the regression coefficients of these five parameters [i.e., the m_i in eq. (11)] were large with high confidence.

3.2 ANALYTICAL METHODS

As described previously, the method used for examining system sensitivity to combinations of parameters found to be most important is to treat each realization of a parameter value as either a "+" or a "−" depending on whether the realized value is greater than or less than a specified value. This is similar to the procedure followed in a signs test[6]. Next, the realizations are sorted based on the commonality of their input parameters being either a "+" or a "−". For example, realizations with all five the important input parameters sampled above the median would be placed in the same bin. Similarly, all realizations where the first four parameters are "+", and the last one is a "−" would be placed in another bin and so on.

For this example, a set of 4,000 realizations of the NRC TPA code was used where 244 input parameters were sampled for the nominal case, which is considered to be the most likely evolution of the repository system.[7] The nominal case includes estimation of WP lifetimes from corrosion and seismically induced rockfall, estimation of radionuclide transport to receptor group locations, and conversion of radionuclide concentration at the receptor location to dose. Table 1 shows some statistical information for these 4,000 realizations. In this introductory example, the branching criterion is the median of the parameter distribution for input variables, and the median of the realized distribution for the output variable (i.e., peak dose in 10 kyr), although other statistical quantities for characterizing the distribution of realizations in a bin are also given. Later in this paper, other branching criteria (e.g., mean, percentiles) will be used with the same data set. Although peak dose for the realization whenever that peak may occur is used in this example, using dose for a realization at the time of the peak of the average dose history curve (i.e., the performance measure in draft versions of the new YM implementing regulation 10 CFR

Part 63[11]) would not significantly alter the results since ~90 percent of the realizations have their peak dose at 10 kyr, and for those realizations with earlier peak doses, the peak dose does not significantly differ from the dose at 10 kyr.

Figure 5 shows the parameter tree based on median values as the branching criterion. In Figure 5, column A is the number of realizations of peak dose that were above the overall median value (i.e., over all 4,000 realizations) in that bin. For example, row one in column A shows that 129 out of 4,000 realizations had all five of the important parameters with values above their median. Of these 129 realizations, 128 had peak doses above the median value for all 4,000 realizations (1.84E-05 rem/yr, Table 1). Column B shows that for these 129 realizations, the mean value of peak dose was 1.20 E-04 rem/yr and column C shows that these 129 realizations accounted for 21.07 percent of the population mean of peak doses. This analysis reinforces the notion that these are indeed important parameters since slightly less than 3 percent of the realizations account for over 21 percent of the mean from all realizations. Column D shows an "importance factor" which is determined as the ratio of the contribution to the overall mean from realizations in that bin to the average contribution of the same number of realizations to the overall mean, i.e.,

$$\begin{aligned} \mathrm{A} &= \frac{\mathit{fractional\ contribution}}{\left(\dfrac{\mathit{number\ of\ realizations\ in\ bin}}{\mathit{total\ number\ of\ realizations}}\right)} \\ &= \frac{\mathit{mean\ peak\ dose\ in\ bin}}{\mathit{mean\ peak\ dose\ over\ all\ realizations}} \end{aligned} \tag{12}$$

All of the data in columns A through D serve as figures of merit for characterizing the group of realizations in a bin. Some other interesting observations that can be made about Figure 5 are:

- The realizations where none or one of the input parameters is "–" account for 67 percent of the mean from all realizations (798 out of 4,000 realizations).

- Only 8 out of 32 bins have importance factors above unity, indicating that the output variable distribution is skewed (the eight bins include 999 out of 4,000 realizations).

In column 2 of Table 2, the sensitivity coefficients calculated based on equations in Section 2 are presented for this example (example 1). The ranking of individual parameters matches the ranking obtained from using the regression analysis with the exception that parameters X_3 and X_4 have been reversed. We emphasize that these sensitivity coefficients provide only the relative sensitivities, so for example, from Table 2, Column 2, one can infer that the system is 1.8 times (0.351/0.192) more sensitive to parameter X_1, than it is to parameter X_5. In the lower portion of Table 2, the system sensitivities to joint sets of parameters are presented. As can be seen in the table, the system shows relatively greater sensitivity to parameter sets of increasing size. Again, one should keep in mind that such results are necessarily dependent on conceptual models embodied in the simulation model as well as on the many fixed value (deterministic) parameters in the TPA code. Other columns of Table 2 pertain to examples described below.

3.3 PARAMETER TREES USING DIFFERENT BRANCHING CRITERIA

As mentioned briefly in the previous section, different branching criteria may be used to determine a "+" or a "–" value for a given parameter or the output variable. Figure 6 shows a tree where both the input parameters and the output variable have been partitioned based on their mean values. Again, the bins toward the top of the tree account for a disproportionate amount of the mean from all 4,000 realizations. In this example, sampling all five of the important input parameters above their mean values assures a peak dose above its mean value (see column A, row 1 in Figure 6). The realizations where none or one of the input parameters is "–" account for 55 percent of the mean from all realizations (528 out of 4,000 realizations) which is a greater fraction on a per realizations basis than the example presented in Figure 5. Column 3 (Example 2) of Table 2 shows that the ranking of the parameters according to sensitivity is slightly different with the mean as the branching criterion; in this case X_2 is the most sensitive parameter. Note however in the last row of table 2, that with the mean as the branching criterion, 99.6 percent of output values exceed its mean when these five parameters take on values greater than their means.

Figure 7 presents an example of the technique where the input parameters are partitioned based on their median values and the output variable is partitioned based on its 90th percentile. Columns B, C, and D of this figure contain numeric entries that are identical to those in Figure 5. Row 1 of column A, however, shows that if all five of the important parameters are sampled above their median values (129 out of 4,000 realizations), then the output variable is above its 90th percentile (4.97E-05 rem/yr) in 103 of these realizations. From Example 3 of table 2, it is apparent that only 79.8 percent of the output above its 90th percentile is provided by the set of five parameters taking on values greater than their median. Comparing to corresponding values for Examples 1 and 2 of Table 2, it is clear that a significant number of extreme values (i.e., above 90th percentile) of the output are produced by combinations of parameters not represented by the group of five used in our examples. Following the stepwise implementation described in the next section, it is possible to determine a set of parameters, different from the above group, that most influence the 90th or other percentiles.

Although these examples use parameter statistics as the branching criteria, other quantities could also be used. For example, total system failure could be defined as a peak dose to the hypothetical receptor greater than a predetermined limit defined by the regulation[11]. Similarly, input parameters could be partitioned based on a value that has some physical significance. For example, in the NRC TPA code, flow in fractures in the UZ is initiated when the infiltration exceeds the saturated matrix conductivity, currently estimated at about 3 mm/yr. This cutoff is important in terms of performance of this subsystem because flow in fractures occurs more rapidly and dissolved contaminants experience much less chemical retardation than flow in the rock matrix. Hence, initiation of fracture flow in the UZ could be thought of as a transition from one performance regime to another for the UZ.

3.4 STEPWISE IMPLEMENTATION OF THE TECHNIQUE

The parameter tree technique was implemented in a stepwise fashion with the importance factor (Column D of Figures 5 through 8) as the figure of merit for determining maximum polarity of the bins and the median value as the branching criterion. First, a one-parameter-depth tree was drawn for each sampled parameter. The parameter that yielded the greatest importance factor for one of the two branches was then used as the first-level parameter for the following iteration in the stepwise implementation. Next, for all remaining sampled parameters, a two-parameter-depth tree was drawn where the first-level parameter was as determined from the previous iteration. In this second iteration, the parameter that yielded the greatest importance factor on any branch of the tree was used as the second-level parameter for the third iteration. The procedure was repeated until the number of realizations in any bin dropped below 50, with the results of that iteration being discarded. This procedure resulted in a tree that was five parameters deep as shown in Figure 8. The reader will note that the first four parameters appear in the same order as in the stepwise regression shown in Figure 4, however, the fifth parameter is the well pumping rate at the receptor location 20 km down gradient- (PR_{20}) instead of the subarea wet fraction (SA_{wf}). This result is important because it shows that these parameters comprise the most important five-parameter set, which differs from the five individually most important parameters as determined by traditional methods (used in the previous example). Also, note that PR_{20} is negatively correlated with the output variable (because in the NRC TPA model increased pumping merely increases the dilution volume and not the interception fraction of the contaminant plume by the well) and the procedure for assigning "+" and "–" was not reversed so the "+ + + + –" bin represents the most pessimistic case in this example (i.e., the bin with the largest peak doses). In figure 8, note that this group of five parameters together produces a higher value of importance factor (7.06) for one of the branches (second branch from top of the tree) as compared to that in figure 5 (6.52 for the topmost branch). In contrast, the sensitivity measures in Table 2 for Example 4 show that the combination of these five parameters (i.e., the last row) have a joint relative sensitivity less than that of Example 1 (26.8 versus 33.5). Thus, the nature of information provided by each sensitivity measure is somewhat different. In other words, if we had decided to implement the stepwise procedure using the joint relative sensitivity measure, the five parameters would match exactly those of Example 1.

3.5 EXAMPLE USING INTERMEDIATE SYSTEM OUTPUTS

As mentioned in a previous section, the technique described in this paper can also be implemented using intermediate system outputs for the branches of the tree as opposed to input parameters. Previous experience indicates that for the HLW repository example, total cumulative radionuclide release (TCR) from the engineered barrier system (EBS), unsaturated zone (UZ), and saturated zone (SZ) are intermediate outputs that are well correlated with peak dose for the compliance period[8] (the output variable), and as such would make excellent examples to use in an "intermediate output" tree. TCR is defined as the integrated release from the subsystem of all radionuclides, measured by activity (Ci), no matter when that release occurs during the compliance period.

Figure 9 shows an intermediate output TCR_{ebs} tree using TCR_{uz} and TCR_{sz} for the branches of the tree. As can be seen in the figure, realizations with all three of these intermediate outputs sampled above their medians dominate performance. In fact, 96.45 percent of the mean value over all realizations is from realizations in this bin. Also, the mean value for realizations in this bin is an order of magnitude higher than in any other bin.

Previously, we did not discuss the issue of correlations between parameters (i.e., between branches of the tree) because these correlations are known *apriori* when the sampling mechanisms in the model are created. However, for the case of intermediate outputs, these correlations are not known. Figure 9 shows that the intermediate outputs considered in this example are correlated. For example, when TCR_{ebs} is greater than its median value or high, TCR_{uz} is also high in 90.4 percent of the realizations.

When TCR_{ebs} and TCR_{uz} are both high, TCR_{sz} is high in 88.5 percent of the realizations. In terms of our previous nomenclature, these probabilities are given as:

$$p(TCR_{uz}+\,|\,TCR_{ebs}+) = \frac{N_{1^+2^+}}{N_{1^+}} \tag{13a}$$

$$p(TCR_{sz}+\,|\,TCR_{uz}+\cap TCR_{ebs}+) = \frac{N_{1^+2^+3^+}}{N_{1^+2^+}} \tag{13b}$$

The magnitudes of these probabilities determine the correlations of the intermediate outputs. The fact that these three subsystem outputs are correlated is expected because in the NRC TPA model, if no release occurs from the EBS, then by definition, no release can occur from the UZ or SZ. Similarly, if no release occurs from the UZ, no release occurs from the SZ. Also, for large EBS releases, the UZ and SZ provide infrequent protection (i.e, they reduce doses by a meaningful amount in only a handful of realizations) which will also lead to correlation of these intermediate outputs. These results, of course, are subject to the assumptions and conceptual models used to generate them and may change as the assumptions and models are updated. The sensitivity coefficients of Section 2 are also calculated and shown in the last column of Table 2.

4 SUMMARY AND CONCLUSIONS

This paper describes a technique for estimating the sensitivity of system output variables to groups of system input parameters via the development of parameter trees. Examples are presented using 4,000 realizations of the NRC TPA code. The technique involves assigning realization-specific parameter values either a "+" or a "–" based on whether or not they are greater or less than a statistic (e.g., median,

31/58

mean, percentile) or some other branching criterion. Similar to event trees, parameter trees are then formed that group realizations into bins (or branches) based on the commonality of their parameter values being either a "+" or a "–". In this manner, groups of parameters can be examined (e.g., realizations where several important parameters are all "+") to determine statistical information about the realizations in a particular bin.

The technique was implemented in a stepwise fashion to determine if the most important (with contribution to the mean as the criterion for importance) five member parameter set is different from the five individually most important parameters determined using traditional sensitivity analysis techniques. It was found that the fifth member of the stepwise constructed set differed from the set constructed with the five individually most important parameters, suggesting that the most sensitive input parameter set may not be the one composed of the individually most sensitive parameters.

The technique was also used with intermediate system outputs on the branches of the tree. It was shown for the NRC TPA code that if total cumulative radionuclide release from the three subsystems (engineered barrier system, unsaturated zone, and saturated zone) were high, then the corresponding peak doses from these realizations dominated total system performance. This tree also clearly showed the strong correlations between these subsystem outputs for this model.

The parameter tree approach of post processing results of *Monte Carlo* simulations provides a useful tool to rank the most important individual parameters and subgroups of parameters from among all subgroups that may be formed. It is relatively simple to implement and provides important insights about the system behavior. Its strength and uniqueness is that it can consider two or more parameters jointly in estimating sensitivities. Its primary weakness is the very large number of realizations required if many parameters are considered.

It is worth reiterating that multiple measures of sensitivity can be defined, each providing different information about system behavior. We defined two in this paper, one based on the number of realizations that meet a specified criterion and the other based on the contribution that a set of parameters makes to the mean of the output. Parameters may differ in their ranking depending upon the criterion used.

ACKNOWLEDGMENTS

This work was performed at the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Nuclear Regulatory Commission (NRC) under Contract No. 02-97-009. The paper is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC.

The authors wish to thank Drs. Gordon Wittmeyer and Wesley Patrick for their thorough reviews of the paper. The authors also benefitted through discussion with Drs. B. John Garrick, Stanley Kaplan and James Lin of the NRC Advisory Committee on Nuclear Waste and with staff members at the NRC, specifically Drs. Keith McConnell and Norman Eisenberg, and Mr. Timothy McCartin and Mr. James Firth.

REFERENCES

1. Draper, N.R.,and H. Smith Jr. 1981. *Applied Regression Analysis- Second Edition.* John Wiley and Sons Publishing. New York, NY.

2. Iman, R.L., J. Davenport, and D. Ziegler. 1980. *Latin Hypercube Sampling (Program User's Guide).* SAND-1473. Sandia National Laboratories. Albuquerque, NM.

3. Helton, J.C., J.W. Garner, R.D. McCurley, and D.K. Rudeen. 1991. *Sensitivity Analysis Techniques and Results for Performance Assessment at the Waste Isolation Pilot Plant.* SAND90-7103. Sandia National Laboratories. Albuquerque, NM.

4. Iman, R.L., and W.J. Connover. 1979. *The Use of the Rank Transform in Regression.* Techtonometrics 21. pp. 499–509.

5. Saltelli, A., J.H. Anderes, and T. Homma. 1993. *Sensitivity Analysis of Model Output—An Investigation of New Techniques.* Elsevier Science Publishers B.V, North Holland. Computational Statistics & Data Analysis 15, pp. 211–238.

6. Bowen, W.M., and C.A. Bennet (eds.). *Statistical Methods for Nuclear Material Management.* NUREG/CR-4604. U.S. Nuclear Regulatory Commission. Washington, D.C.

7. Mohanty, S., and T.J. McCartin (eds.). 1998. *NRC Sensitivity and Uncertainty Analyses for a Proposed HLW Repository at Yucca Mountain, Nevada Using TPA 3.1- Volume I: Conceptual Models and Data.* NUREG-1668. U.S. Nuclear Regulatory Commission. Washington, D.C.

8. Jarzemba, M.S., R. Codell, L. Deere, J. Firth, C. Lui, S. Mohanty, K. Poor, and J. Weldy. *NRC Sensitivity and Uncertainty Analyses for a Proposed HLW Repository at Yucca Mountain, Nevada Using TPA 3.1- Volume II: Results and Conclusions.* NUREG-1668. U.S. Nuclear Regulatory Commission. Washington, D.C.

9. StatSci. 1993. *S-PLUS User's Manual- Version 3.2.* MathSoft, Inc. Seattle, WA.

10. Wescott, R.G., M.P. Lee, N.A. Eisenberg, T.J. McCartin, and R.G. Baca (eds.). 1995. *NRC Iterative Performance Assessment Phase 2-Development of Capabilities for Review of a Performance Assessment for a High-Level Waste Repository*. NUREG-1464. U.S. Nuclear Regulatory Commission. Washington, D.C.

11. U.S. Nuclear Regulatory Commission. 1999. Disposal of High-Level Radioactive Wastes in a Proposed Geological Repository at Yucca Mountain, Nevada; Proposed Rule. Federal Register. Vol. 64, No. 34. pp. 8,640–8,679.

FIGURE CAPTIONS

1. A general parameter tree.
2. A diagram showing the breakdown of a total system into subsystem components and their intermediate outputs.
3. A diagram illustrating the U.S. Nuclear Regulatory Commission performance assessment model.
4. Results of the stepwise multilinear regression showing the parameters with most influence on residual sum of squares.
5. A tree describing the new technique for examining system sensitivity to groups of parameters. Input parameters divided based on their median values and output variable divided based on its median value from all 4,000 realizations.
6. Another tree describing the new technique for examining system sensitivity to groups of parameters. Input parameters divided based on their mean values and output variable divided based on its mean value from all 4,000 realizations.
7. Another tree describing the new technique for examining system sensitivity to groups of parameters. Input parameters divided based on their median values and output variable divided based on its 90th percentile value from all 4,000 realizations.
8. A tree developed using a stepwise implementation of the technique based on the importance factor.
9. An intermediate output tree developed using total cumulative radionuclide release from the engineered barrier system, unsaturated zone, and saturated zone based on 1,440 realizations of the TPA code.

34/58

Table 1. Statistical information about the 4,000 realizations.

Parameter	Median Value	Mean Value	90th Percentile	Distribution Type [bounds]
I_0 (mm/yr)	5.5	5.5	9.1	Uniform [1,10]
F_{ow}	0.173	0.264	0.566	Lognormal [0.1,3.0]
WP_{def}	0.00505	0.00505	0.00901	Uniform [0.0001,0.01]
F_{mult}	0.0447	0.0503	0.0833	Lognormal [0.01,0.2]
SA_{wf}	0.5	0.5	0.9	Uniform [0.0,1.0]
realization peak dose (rem/yr)	2.82E-06	1.84E-05	4.97E-05	—

Table 2. Sensitivity coefficients for the example trees in this paper.

	Coefficient	Example 1: Figure 5	Example 2: Figure 6	Example 3: Figure 7	Example 4: Figure 8	Example 5: Figure 9
Unconditional Sensitivities of Individual Parameters	S_{X_1}	0.351	0.26	0.134	0.351	0.772
	S_{X_2}	0.31	0.28	0.16	0.31	0.692
	S_{X_3}	0.202	0.173	0.119	0.202	0.192
	S_{X_4}	0.204	0.178	0.0839	0.204	N/A
	S_{X_5}	0.192	0.15	0.102	0.081	N/A
Joint Sensitivities of Parameter Groups	$\frac{\lvert p_H - p_L \rvert}{1-\lvert p_H - p_L \rvert}$	0.541	0.351	0.155	0.541	3.39
		2.37	1.63	0.462	2.37	9.53
		6.75	4.68	0.938	6.75	19.8
		26.0	9.42	1.46	26.0	N/A
		33.5	249	3.95	26.8	N/A

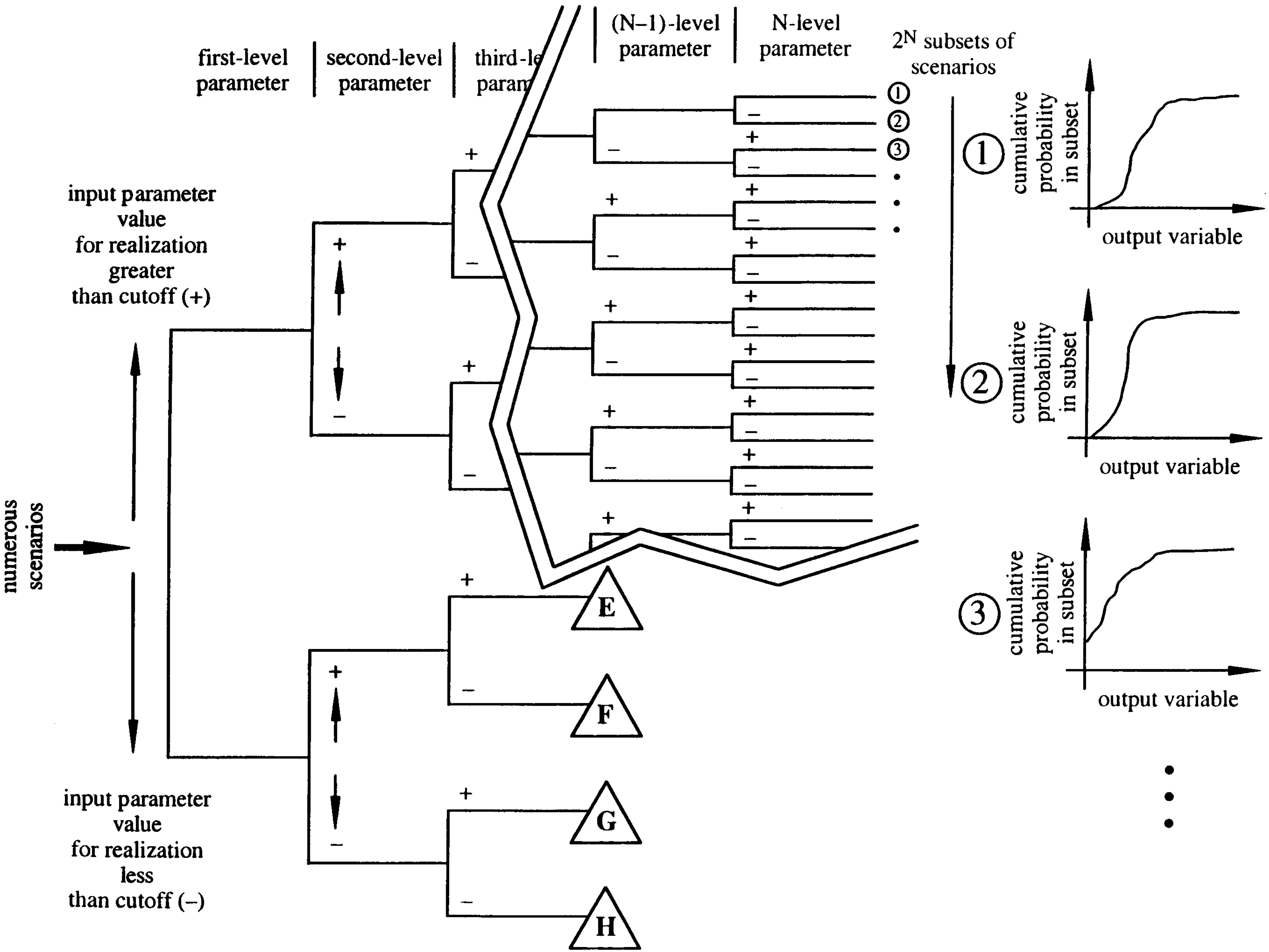

first-level parameter
second-level parameter
third-level parameter
(N–1)-level parameter
N-level parameter
2^N subsets of scenarios
numerous scenarios
input parameter value for realization greater than cutoff (+)
input parameter value for realization less than cutoff (–)
+
–
①
②
③
E
F
G
H
cumulative probability in subset
output variable

Figure 1

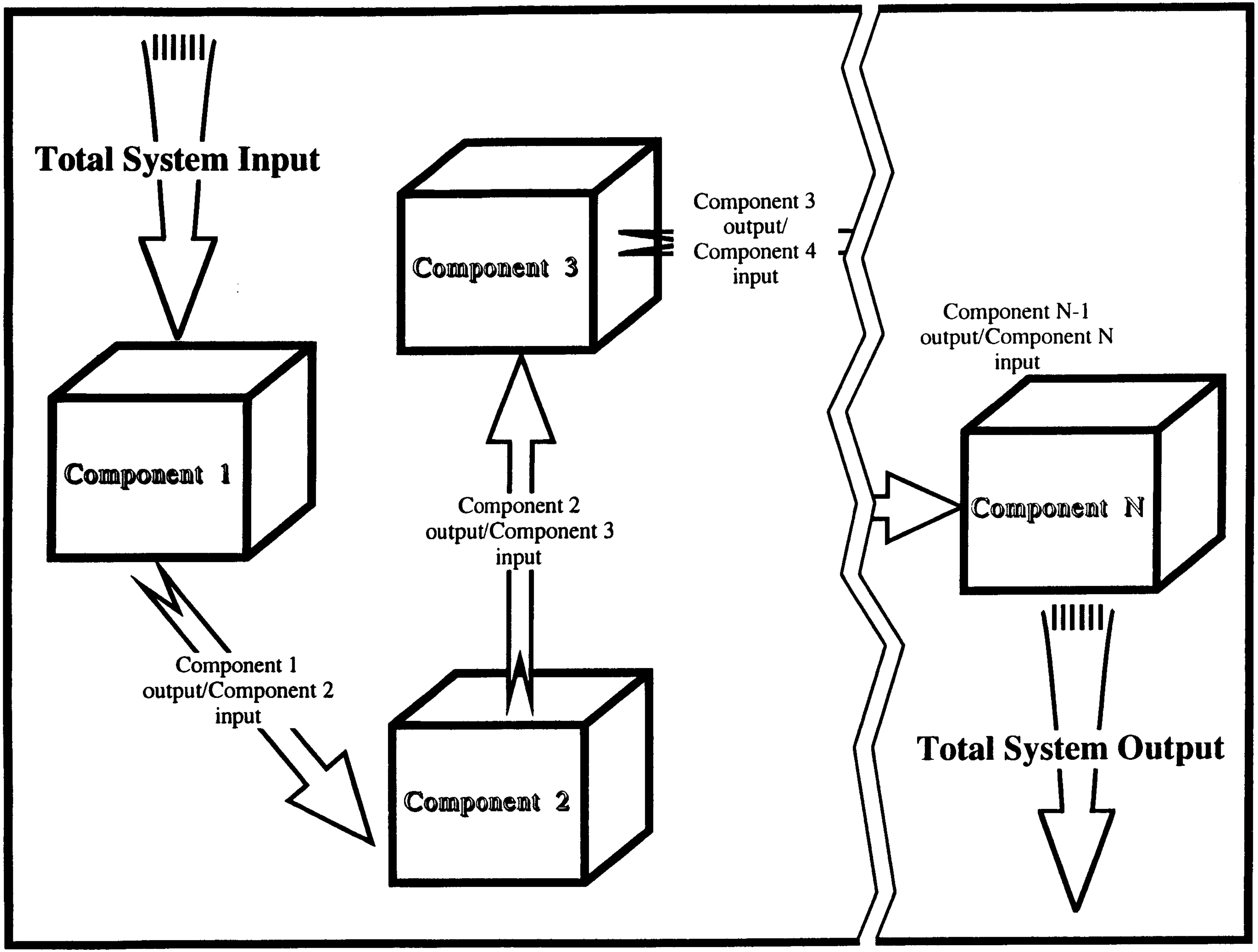

Total System Input
Component 3
Component 3 output/ Component 4 input
Component N-1 output/Component N input
Component 1
Component 2 output/Component 3 input
Component N
Component 1 output/Component 2 input
Component 2
Total System Output

Figure 2

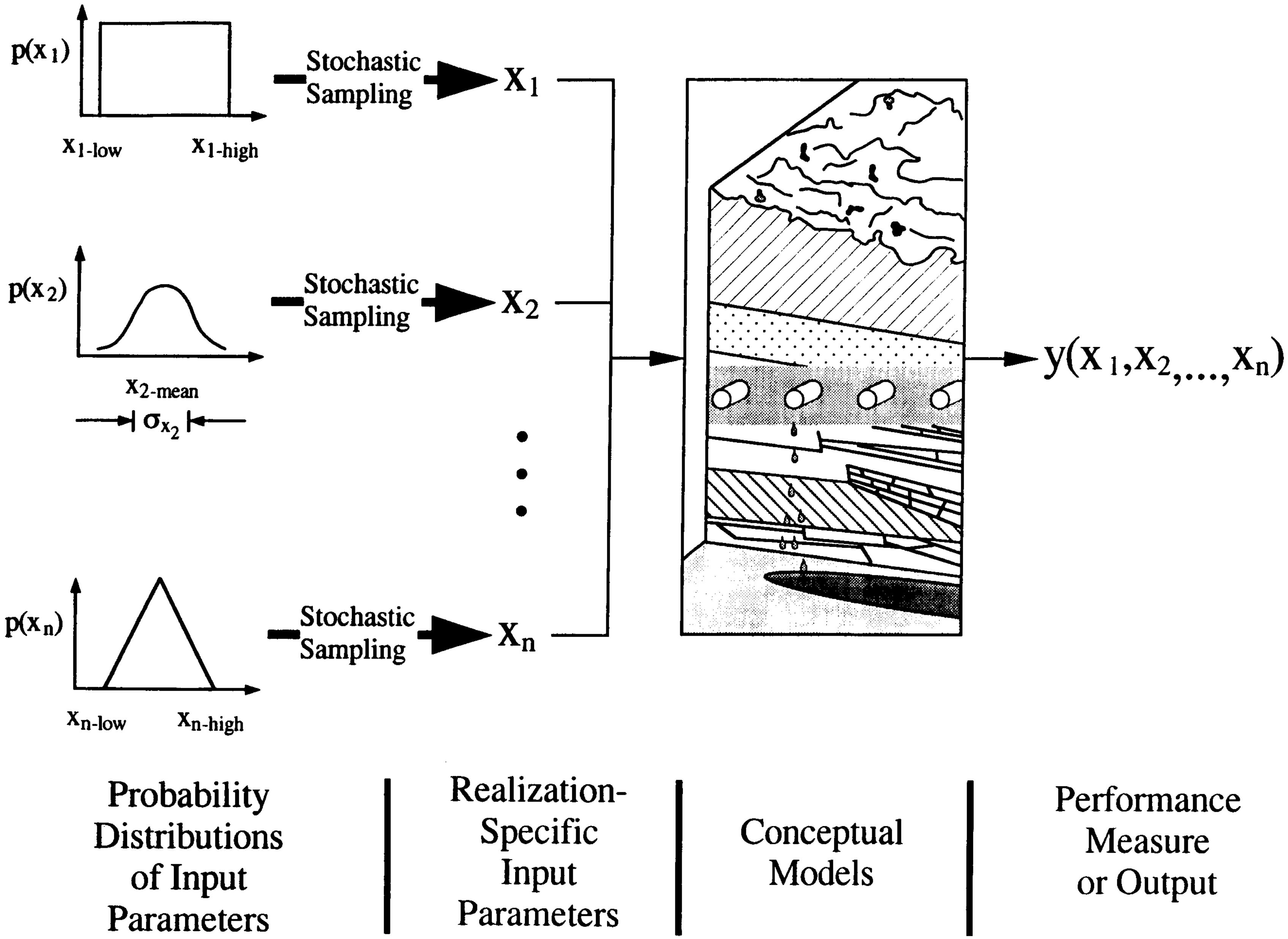
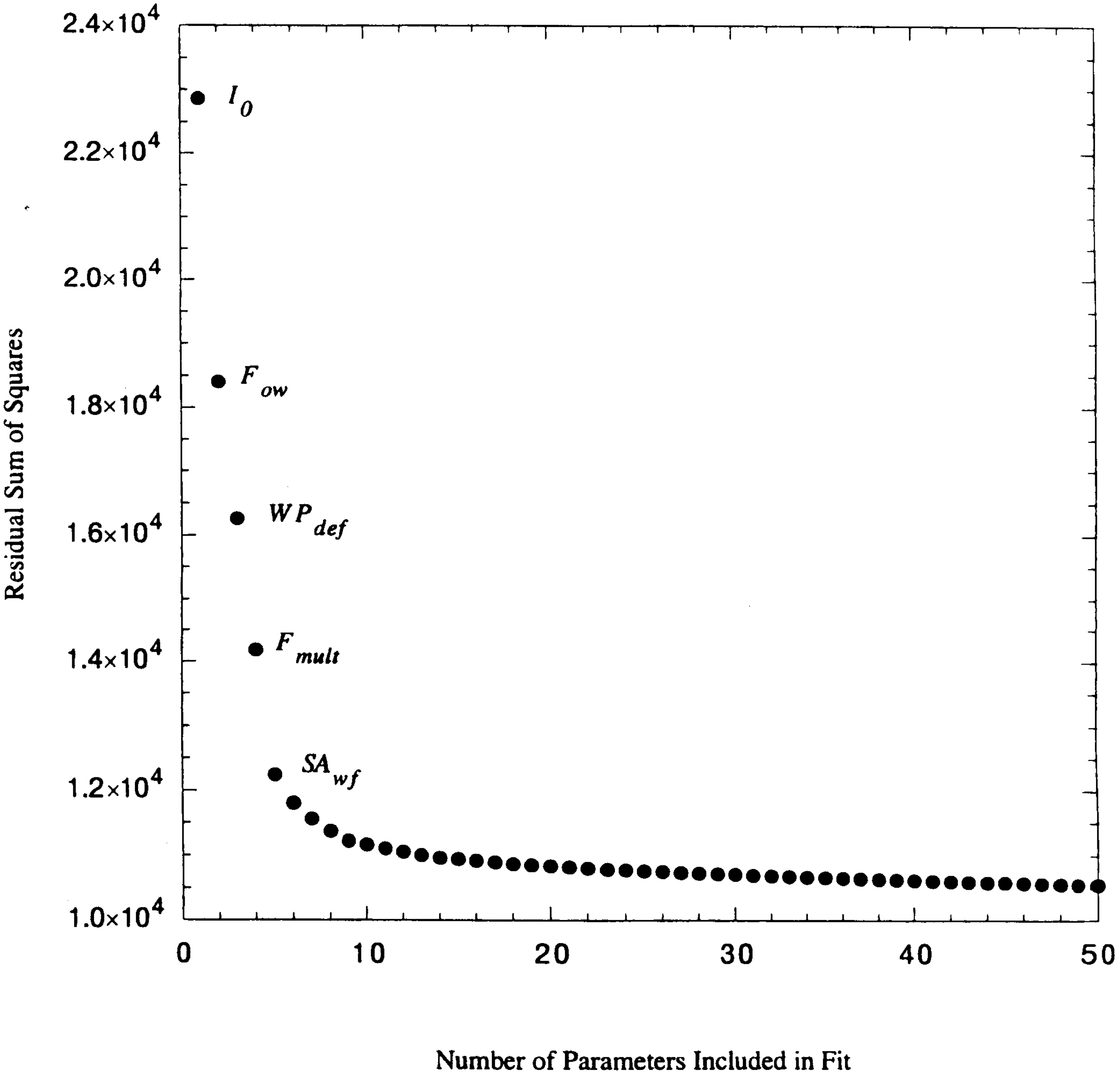
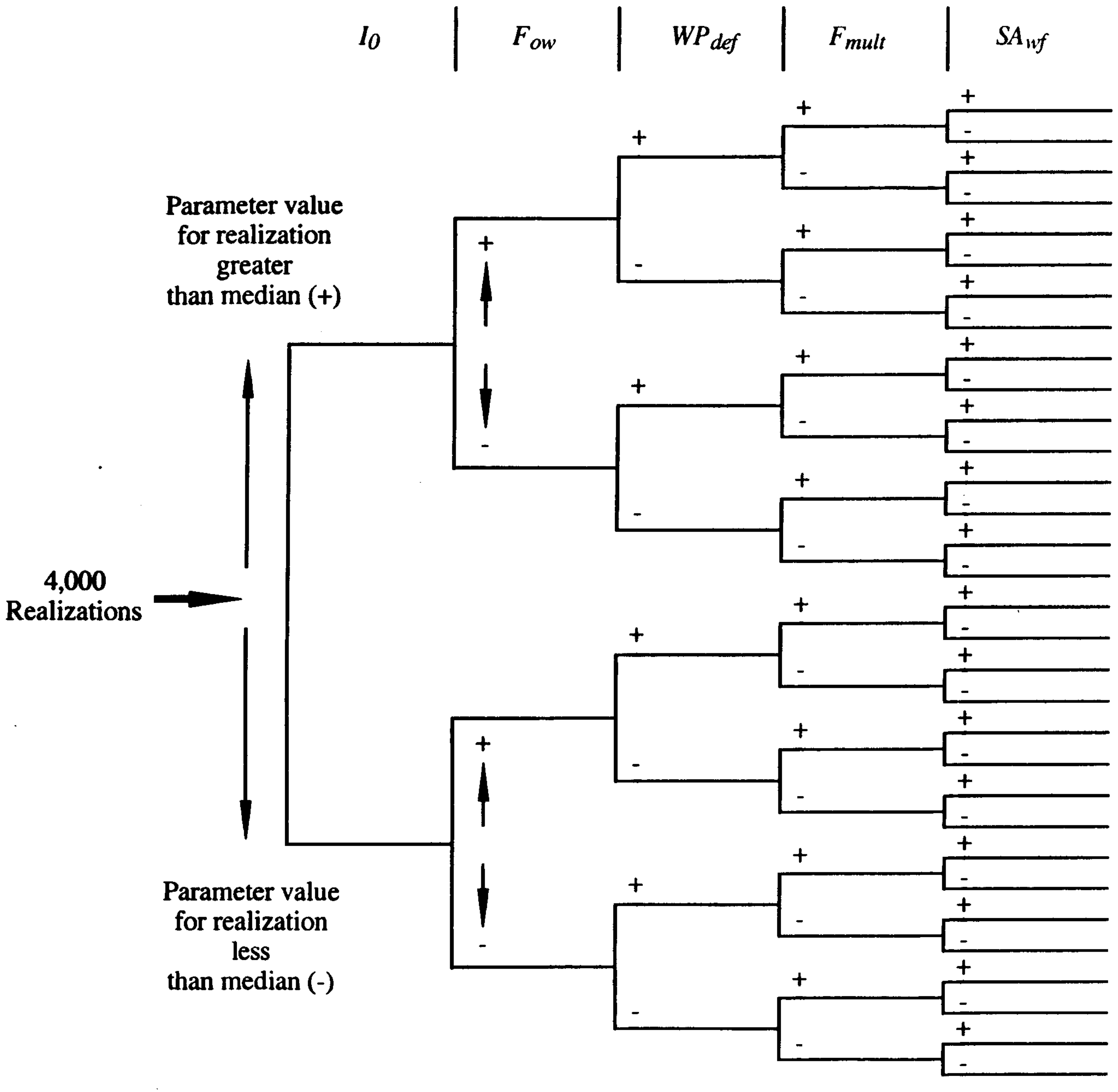

$p(x_1)$
$x_{1\text{-low}}$
$x_{1\text{-high}}$
Stochastic Sampling
x_1
$p(x_2)$
$x_{2\text{-mean}}$
σ_{x_2}
Stochastic Sampling
x_2
$p(x_n)$
$x_{n\text{-low}}$
$x_{n\text{-high}}$
Stochastic Sampling
x_n
$y(x_1, x_2, \ldots, x_n)$
Probability Distributions of Input Parameters
Realization-Specific Input Parameters
Conceptual Models
Performance Measure or Output

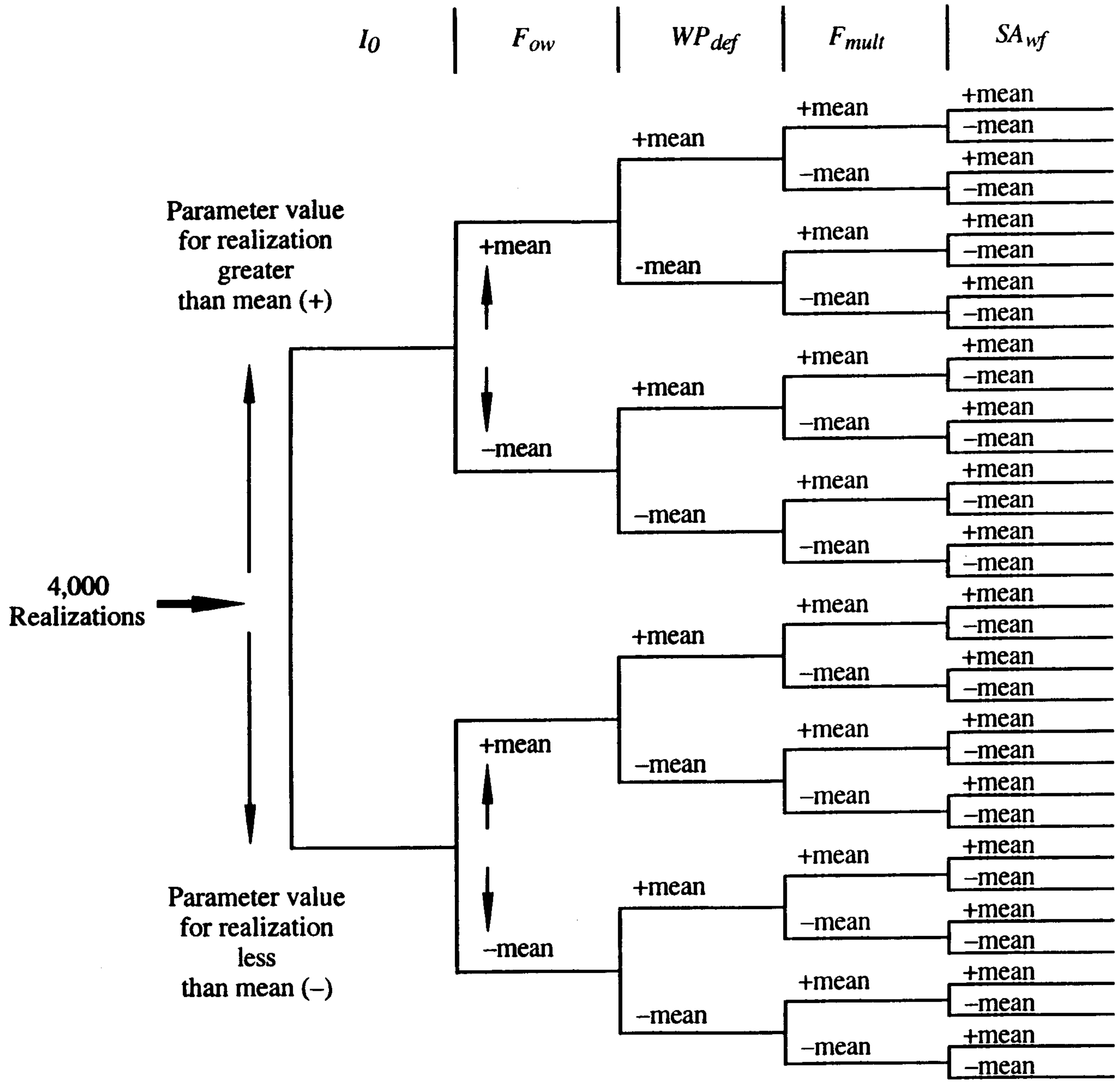
Figure 3


2.4×10^4
2.2×10^4
2.0×10^4
1.8×10^4
1.6×10^4
1.4×10^4
1.2×10^4
1.0×10^4
Residual Sum of Squares
I_0
F_{ow}
WP_{def}
F_{mult}
SA_{wf}
0
10
20
30
40
50
Number of Parameters Included in Fit



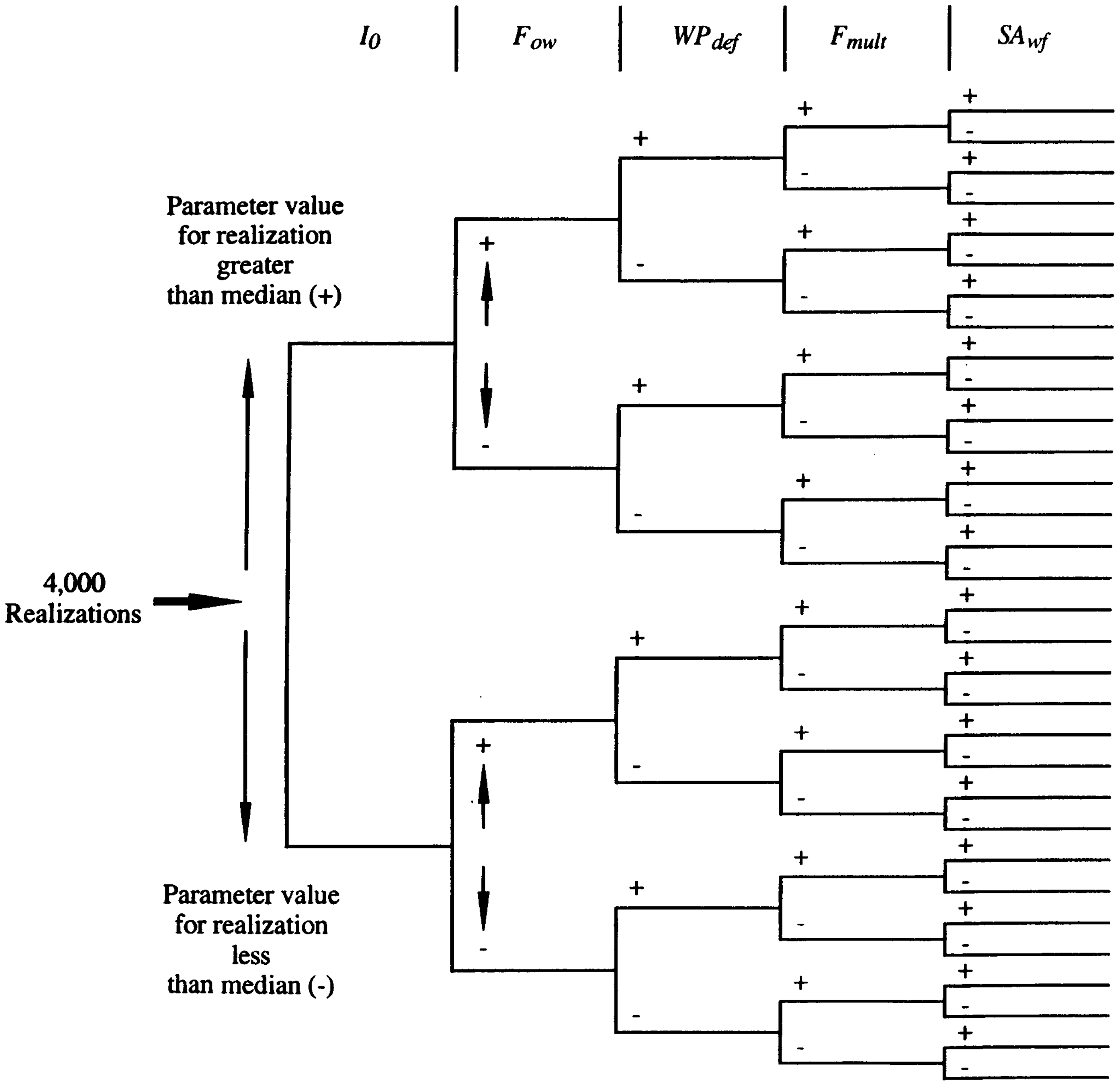
A	B	C	D
128/129	1.20E-04	0.2107	6.52
96/97	6.11E-05	0.0803	3.32
135/140	6.64E-05	0.1261	3.61
57/68	2.27E-05	0.0209	1.23
130/141	4.68E-05	0.0894	2.54
53/70	1.52E-05	0.0114	0.83
104/133	2.19E-05	0.0395	1.19
49/87	9.63E-06	0.0114	0.52
179/224	4.04E-05	0.1226	2.20
49/58	1.78E-05	0.0140	0.97
128/216	9.78E-06	0.0286	0.53
20/59	3.73E-06	0.0030	0.20
135/223	1.05E-05	0.0317	0.57
20/62	4.19E-06	0.0035	0.23
62/230	3.19E-06	0.0099	0.17
6/63	1.05E-06	0.0009	0.06
59/67	4.34E-05	0.0395	2.36
125/226	1.72E-05	0.0528	0.93
35/53	1.79E-05	0.0129	0.97
85/230	5.86E-06	0.0186	0.33
42/60	1.18E-05	0.0096	0.64
91/227	6.11E-06	0.0188	0.33
29/54	1.29E-05	0.0094	0.70
39/218	2.19E-06	0.0065	0.12
37/73	8.52E-06	0.0084	0.46
35/125	4.00E-06	0.0068	0.22
22/87	1.90E-06	0.0022	0.10
12/148	7.60E-07	0.0015	0.04
21/96	2.23E-06	0.0029	0.12
11/122	1.03E-06	0.0017	0.06
3/74	6.33E-07	0.0006	0.03
3/140	4.47E-07	0.0008	0.02

Figure 5



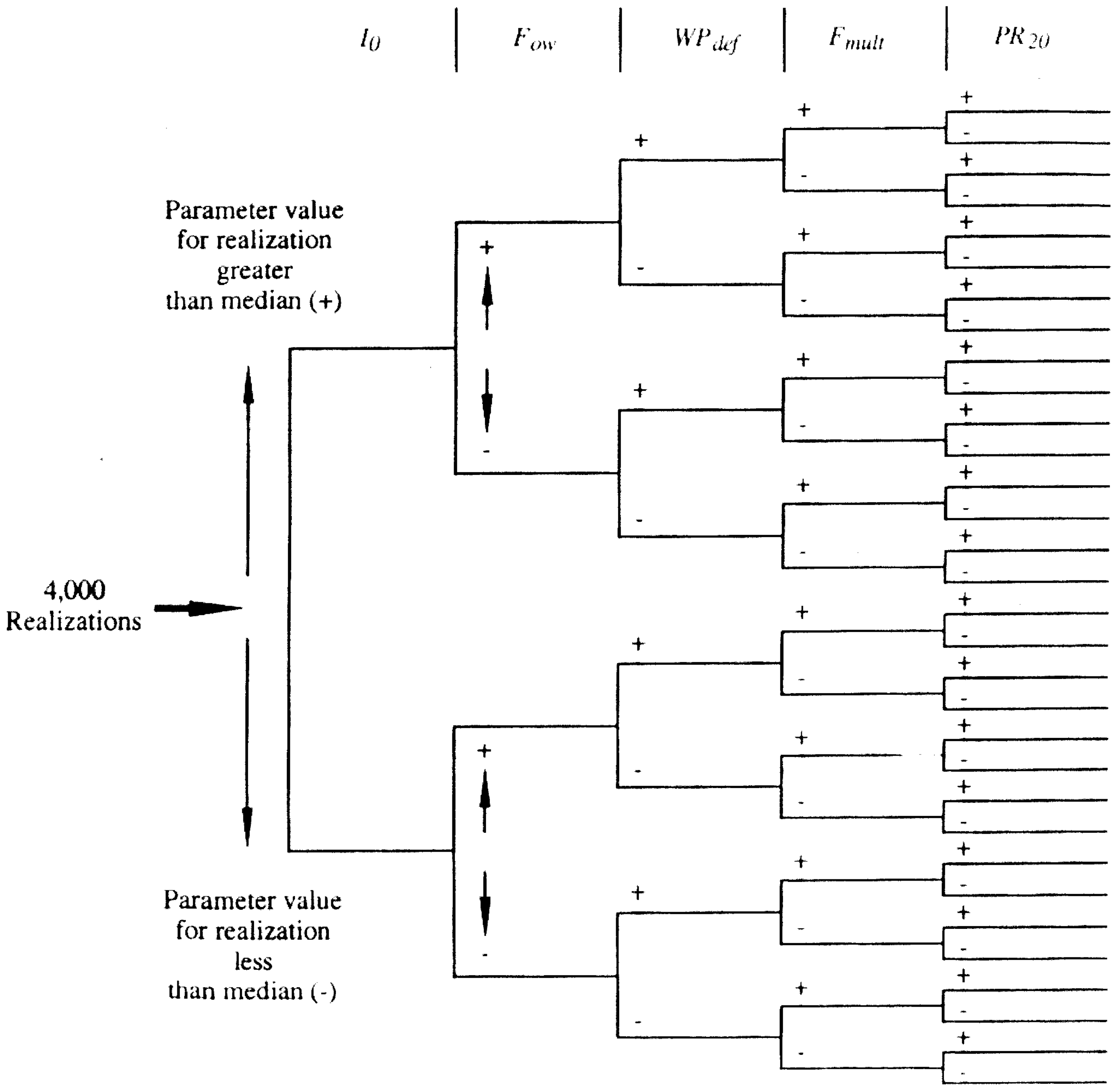
A	B	C	D
62/62	1.48E-04	0.1244	8.04
42/52	8.31E-05	0.0586	4.52
85/96	9.94E-05	0.1294	5.40
34/53	3.66E-05	0.0263	1.99
48/61	4.94E-05	0.0408	2.68
13/35	1.86E-05	0.0088	1.01
55/94	3.29E-05	0.0420	1.79
17/78	1.21E-05	0.0128	0.66
128/229	5.42E-05	0.1683	2.95
41/72	2.31E-05	0.0226	1.26
82/322	1.51E-05	0.0659	0.82
12/105	7.50E-06	0.0107	0.41
66/220	1.95E-05	0.0581	1.06
8/73	6.11E-06	0.0060	0.33
35/352	6.20E-06	0.0296	0.34
0/96	1.95E-06	0.0025	0.11
23/28	6.30E-05	0.0239	3.42
47/138	2.14E-05	0.0401	1.16
17/39	2.44E-05	0.0129	1.33
20/185	8.05E-06	0.0202	0.44
7/27	1.45E-05	0.0053	0.79
14/120	7.58E-06	0.0123	0.41
10/41	1.73E-05	0.0096	0.94
9/185	3.35E-06	0.0084	0.18
25/90	1.77E-05	0.0217	0.96
18/157	6.69E-06	0.0142	0.36
1/123	2.71E-06	0.0045	0.15
3/249	1.52E-06	0.0051	0.08
7/96	4.34E-06	0.0056	0.24
5/159	2.30E-06	0.0050	0.13
2/120	1.23E-06	0.0020	0.07
1/243	6.47E-07	0.0021	0.04

Figure 6

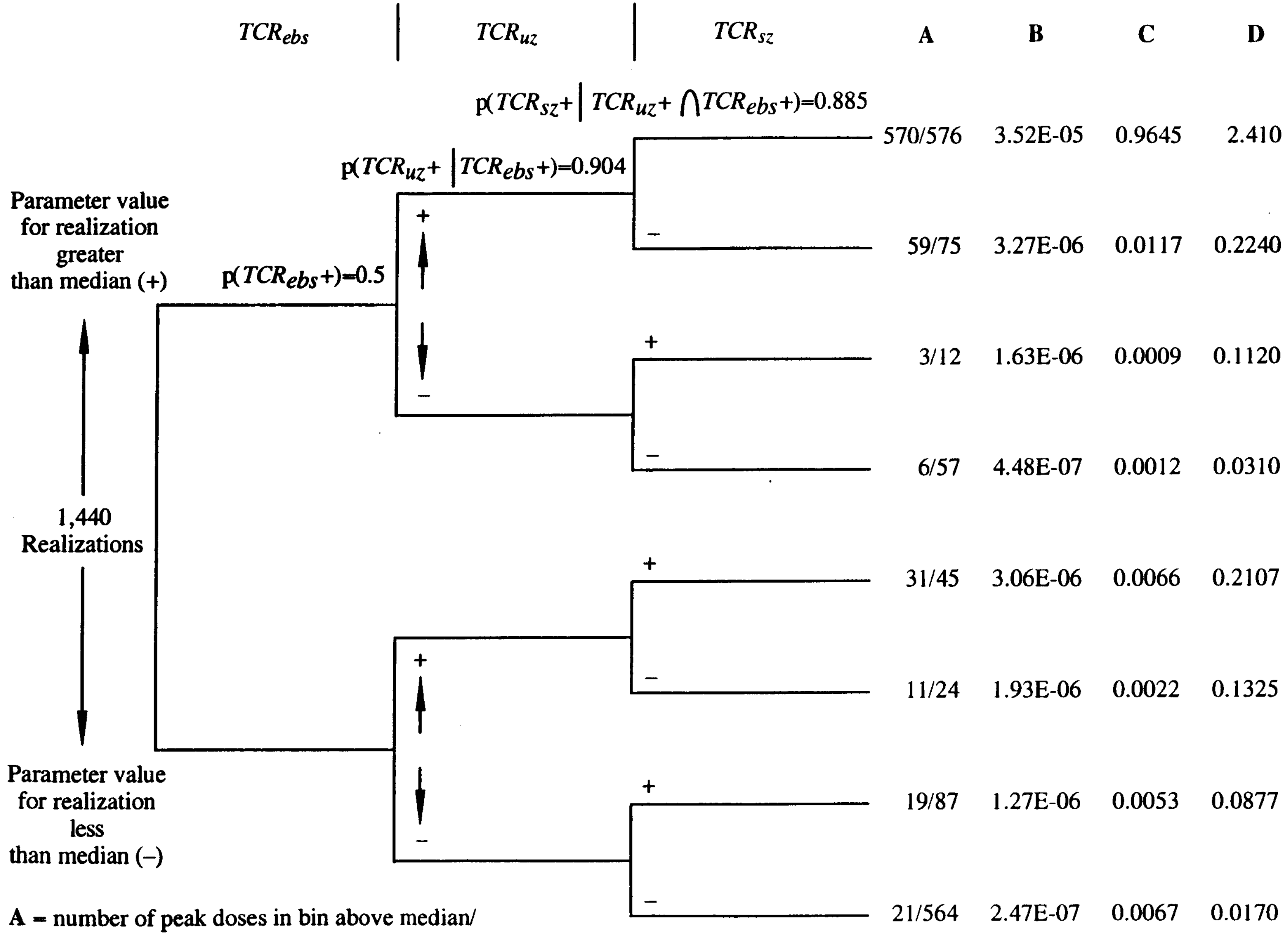


A	B	C	D
103/129	1.20E-04	0.2107	6.52
31/97	6.11E-05	0.0803	3.32
65/140	6.64E-05	0.1261	3.61
11/68	2.27E-05	0.0209	1.23
44/141	4.68E-05	0.0894	2.54
4/70	1.52E-05	0.0114	0.83
16/133	2.19E-05	0.0395	1.19
3/87	9.63E-06	0.0114	0.52
41/224	4.04E-05	0.1226	2.20
4/58	1.78E-05	0.0140	0.97
6/216	9.78E-06	0.0286	0.53
0/59	3.73E-06	0.0030	0.20
6/223	1.05E-05	0.0317	0.57
0/62	4.19E-06	0.0035	0.23
0/230	3.19E-06	0.0099	0.17
0/63	1.05E-06	0.0009	0.06
21/67	4.34E-05	0.0395	2.36
25/226	1.72E-05	0.0528	0.93
5/53	1.79E-05	0.0129	0.97
3/230	5.86E-06	0.0186	0.33
1/60	1.18E-05	0.0096	0.64
2/227	6.11E-06	0.0188	0.33
5/54	1.29E-05	0.0094	0.70
0/218	2.19E-06	0.0065	0.12
3/73	8.52E-06	0.0084	0.46
1/125	4.00E-06	0.0068	0.22
0/87	1.90E-06	0.0022	0.10
0/148	7.60E-07	0.0015	0.04
0/96	2.23E-06	0.0029	0.12
0/122	1.03E-06	0.0017	0.06
0/74	6.33E-07	0.0006	0.03
0/140	4.47E-07	0.0008	0.02

Figure 7



A	B	C	D
124/124	6.59E-05	0.1109	3.58
100/102	1.30E-04	0.1801	7.06
82/90	4.31E-05	0.0526	2.37
110/118	5.90E-05	0.0944	3.20
84/102	2.81E-05	0.0389	1.53
99/109	4.39E-05	0.0649	2.38
74/109	1.44E-05	0.0213	0.78
79/111	1.96E-05	0.0295	1.06
102/137	1.86E-05	0.0345	1.01
126/145	5.19E-05	0.1021	2.82
66/134	5.51E-06	0.0100	0.30
82/141	1.13E-05	0.0216	0.61
66/135	6.78E-06	0.0124	0.37
89/150	1.12E-05	0.0228	0.61
23/145	1.71E-06	0.0034	0.09
45/148	3.73E-06	0.0075	0.20
84/153	1.55E-05	0.0321	0.84
100/140	3.17E-05	0.0601	1.72
57/145	6.93E-06	0.0136	0.38
63/138	9.55E-06	0.0179	0.52
54/133	5.69E-06	0.0103	0.31
79/154	8.67E-06	0.0181	0.47
29/148	2.97E-06	0.0060	0.16
39/124	5.92E-06	0.0100	0.32
30/95	4.11E-06	0.0053	0.22
42/103	7.10E-06	0.0099	0.39
17/115	1.23E-06	0.0019	0.67
17/120	1.13E-06	0.0018	0.06
17/111	1.53E-06	0.0023	0.08
15/107	1.60E-06	0.0023	0.09
2/124	3.23E-07	0.0005	0.02
4/90	7.70E-07	0.0009	0.04



A = number of peak doses in bin above median/ number of realiztions in bin
B = mean peak dose in bin (rem/yr)
C = fractional contribution of bin to overall mean
D = importance factor

Figure 9

SOFTWARE DEVELOPMENT PLAN - SUPERMODSIGN POST-PROCESSOR FOR TPA VERSION 3.2

by

Kevin Poor

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

Reviewed by:

James R. Weldy 4/8/99
James Weldy

Approved by:

Gordon Wittmeyer 4/8/99
Gordon Wittmeyer, Element Manager, Performance Assessment

SOFTWARE DEVELOPMENT PLAN - SUPERMODSIGN POST-PROCESSOR FOR TPA VERSION 3.2

1 SCOPE

Supermodsign is a post-processor tool for developing parameter trees from Total-system Performance Assessment (TPA) Version 3.2 code results. Since parameter trees are similar to event trees, which are commonly used to present risk assessment results for other complicated systems (e.g., nuclear reactors), it is expected that their application to repository performance assessment (PA) would help make the results of complex PA models more transparent. The application will be written in standard FORTRAN90 and hence it will have no platform- or system-specific requirements. A brief users guide will accompany the software.

2 BASELINE ITEMS

- Parameter tree post processor for TPA version 3.2 output
- Code to read the data files: reads and formats data from a TPA run
- Test Data: sample TPA run data used to test the program
- Users Guide: describes computational algorithms, system requirements, and code usage procedures

3 PROJECT MANAGEMENT

3.1 PROJECT SCHEDULE

Work will begin in late-March 1999 and continue through the end of June 1999.

3.2 STAFFING

One CNWRA staff member working part time with limited contractor support have been assigned for the duration of the project.

3.3 RISK MANAGEMENT

The current scope of this project is to develop a stand-alone code as described in Jarzemba and Sagar (1999). Additional development may be required in the future to incorporate the Supermodsign code into existing TPA post processors. Upcoming changes in CNWRA staff availability may affect code development. To allow completion of code development and testing on schedule, additional PA element subcontractor support is being considered.

3.4 WORK BREAKDOWN STRUCTURE

All work will be completed under the TPA code development component of the work breakdown structure (i.e., 20-1402-762). Required labor is estimated in the following table.

Task	Estimated Labor Hours
Generate preliminary system requirements	10
Develop code to read TPA output files	20
Test and refine data read capability	10
Develop code to generate parameter trees and analysis results for user defined TPA input parameters	100
Develop code to serially evaluate (i.e., in a stepwise manner) all stochastic TPA input parameters to develop parameter tree based on an importance factor	80
Test and refine parameter tree development and analysis capability	80
Test cross-platform portability	10
Produce User's Guide	40
Modify code based on customer feedback	40
Prepare project management administrative paperwork	10
Total	400

4 DEVELOPMENT PROCEDURES

4.1 HARDWARE AND SOFTWARE RESOURCES

The software will be developed on a Pentium-based PC running Windows NT. It will be ported to and tested on a Pentium-based PC running Windows 95/98 and a Sun work station running Solaris. These hardware resources are already available in-house and do not need to be purchased.

4.2 SOFTWARE DEVELOPMENT LIFECYCLE

The following describes events in the development of the herein described software:

Analysis—determine input data format, formulate requirements for interface, determine output requirements and format.

Phase I product development—develop code input and output capability, develop parameter tree and analysis algorithms for user-defined TPA stochastic input parameters.

Phase I product testing—test phase I product parameter tree development and analysis algorithms with commercially available spreadsheet sorting routines. Additionally, test program data input and output capabilities.

Phase II product development—develop Supermodsign code to allow multiple level parameter tree development and identification of those input parameters that most affect output response by sequential (i.e., stepwise) analysis of all TPA input parameters.

Phase II product testing—test phase II product parameter tree development and analysis algorithms with commercially available spreadsheet sorting routines.

Iteration release—developers release a version of the software to users.

Testing and user feedback—users provide developers feedback on "look and feel" and functionality of product. Developer uses this information to develop final version of software.

Final delivery—developers provide final version of software to users.

4.3 CODING

The Supermodsign TPA post processor will be written in standard Fortran 90.

4.4 ACCEPTANCE TESTING AND ANALYSIS

SwRI or subcontractor personnel will perform preliminary acceptance testing on the Supermodsign software. Testers will document results of testing and proposed changes using accepted CNWRA Quality Assurance methods (e.g., scientific notebook). The code developers will use this information to make revisions to the program. After preliminary acceptance testing, the end user (the U.S. Nuclear Regulatory Commission) will use the software and provide comments and change requests to the CNWRA, so that the code can be further modified.

5 CONFIGURATION MANAGEMENT PLAN

5.1 TOOLS

Due to the relatively small size of the program, no special configuration management tools are required for this project.

5.2 CONFIGURATION IDENTIFICATION

The code to develop and analyze parameter trees, the code test data, and the User's Guide will be placed under configuration control after customer acceptance. In order to place a particular release version under configuration control, the developers will create a folder named Supermodsign*mmdd*, where *mmdd* is the date the folder was created. This folder will be archived.

5.3 CONFIGURATION PROCEDURES

Due to the small size of the Supermodsign program and development staff, no check-in/check-out procedures are required during the code development and testing phase. Release versions of the Supermodsign software will be cleared through and approved by CNWRA personnel. No official documentation such as an SCR is required for changes to the software during preliminary acceptance testing. Once the code is baselined and ready for delivery to the end user, the end user may request changes to the software using an SCR.

6 REFERENCES

Jarzemba, M.S., and B. Sagar. 1999. A Feasibility Study for a TPA Version 3.2 Event-Tree Post Processor. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

45/58

To: TPA 3.2 PP
Software
Folder

INSTALLATION AND EXECUTION OF THE POST-PROCESSOR FOR THE TOTAL-SYSTEM PERFORMANCE ASSESSMENT (TPA) VERSION 3.2 CODE

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-97-009

Prepared by

Hollis Thomas
Sitakanta Mohanty
Robert Rice
Mario Montoya

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

December 1998

ACKNOWLEDGMENTS

This work was performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards, Division of Waste Management under contract No. NRC-02-97-009. This document is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC.

The TPA 3.2PPβ code has been developed following the procedures described in the CNWRA Technical Operating Procedure, TOP-018, which implements the QA guidance contained in the CNWRA QA Manual.

The authors thank Gordon Wittmeyer and Wesley Patrick for their review of this reports. The authors are thankful to Cathy Garcia for her typing and formatting help in preparing the document.

47/58

CONTENTS

48/58

FIGURES

1 INTRODUCTION

The Total-system Performance Assessment (TPA) Version 3.2 code was recently developed by CNWRA and NRC staff for conducting performance assessments of the proposed high-level radioactive waste repository at Yucca Mountain (YM). Because of the volume and variety of data produced by the various modules in the TPA code, it is cumbersome for a code user to quickly review output from a TPA run. A postprocessor has been developed to plot a variety of standard graphs from data generated by the TPA code.

The postprocessor is written in the Java and FORTRAN programming languages and runs on platforms having either UNIX or MS Windows operating systems. The program consists of an intuitive and easy-to-use interface that allows a user to select and display a plot. Since Java was developed to be used on multiple platforms without modifying source code, the postprocessor will execute on any machine that hosts a Java virtual machine. However, discussions in this report are limited to the operating systems and hardware on which the postprocessor was tested.

The postprocessor has been developed specifically for processing outputs from the TPA Version 3.2 code and, thus, has been designated as TPA3.2PPβ. However, the processor will work for any future version of the TPA code as long as the internal structure of the TPA output files is not changed. On-line help displayed by the Java postprocessor provides details for the graphs, TPA code output files from which data are used in these graphs, and the processing done to the data before the graphs are displayed.

2 OVERALL STRUCTURE

Plotting TPA outputs involves a two-step process. The first step includes processing the TPA output files using the FORTRAN code *fort_process.f*. This code reads data from the *tpa.inp* input file and TPA output from files with *.res* and *.tpa* extensions. The *fort_process.f* code manipulates the data and writes files with *.plt* extensions. The Java postprocessor reads the *.plt* files and generates plots. The *fort_process.f* code is quite general and allows the user to specify the nuclide, realization, and subarea from which to generate data for the Java postprocessor in *fort_process.inp*. In addition, the recurrence rate probabilities for faulting and volcanic disruptive events can be changed in the *fort_process.f* input file. Although this flexibility exists, changes other than to the recurrence rate probability are not compatible with the Java postprocessor buttons and the graph titles in the current version.

Data can be plotted at two time periods: (i) the maximum simulation period and (ii) the compliance period. There are fifteen plots corresponding to each period. Descriptions of these plots are presented in section 6.

A script file, *tparun*, allows the user to execute the TPA code and generate the *.plt* files, which are then read by the Java processor, using the *fort_process.f* FORTRAN code.

3 SYSTEM REQUIREMENTS

3.1 SOFTWARE REQUIREMENTS

The following system requirements are based on the operating systems that were used during the testing phase:

- Windows 95, Windows 98, Windows NT (4.0), or UNIX operating system
- Java Development Kit (JDK 1.1.7 or greater). The JDK is available free on the Internet at `www.java.sun.com`
- TPA Plotting Tool classes (included with software delivery)
- JClass Chart classes (included with the software delivery)
- Data files generated by the TPA code are also required, although sample data files are included with the delivery

3.2 HARDWARE REQUIREMENTS

The following minimum hardware requirements are based on the experience gained during the testing phase of the code:

- Pentium processor (166 MHz or faster) or equivalent UNIX machine
- 32MB RAM
- 30MB hard disk space

4 CODE INSTALLATION AND EXECUTION

A step-by-step procedure for installation and execution of the postprocessor codes is described in this section.

4.1 FORTRAN PROCESSOR

The FORTRAN code *fort_process.f* has been designed to run on a UNIX platform and has software and hardware requirements identical to the TPA Version 3.2 code. A full description of the requirements can be found in the User's Guide for the TPA Version 3.2 code (Mohanty and McCartin, 1998). The following steps are used to install and execute the FORTRAN processor.

- Copy files *fort_process.f*, *fort_process.inp*, and *tparun* from the CD ROM or the diskette in which the source code is provided to the TPA code directory level where the files *tpa.inp* and *tpa.e* reside.

- Compile *fort_process.f* by typing `f77 fort_process.f -o fort_process.e.`
- If output files from the TPA code already exist, the user can generate the *.plt* files simply by typing *fort_process.e*. Otherwise the user must type *tparun* to generate TPA code outputs and *.plt* files.

4.2 JAVA PROCESSOR

The Java processor has been tested on PC platforms running the Windows NT 4.0, Windows 95, and Windows 98 operating systems and on a UNIX platform (Scratchy1: SUN Sparc20 with Solaris 2.5.1 Operating System). Therefore, instructions are provided only for these platforms and operating systems, although the Java processor is presumed to be platform-independent. The following steps should be followed to install and execute the Java processor.

4.2.1 Windows NT

- Copy the Plotter folder to `c:\Plotter`
- Install JDK version 1.1.7 to `c:\jdk1.1.7` by following the JDK installation instructions. Get JDK at www.java.sun.com.
- Copy `jcchart300.jar` to `c:\jdk1.1.7\lib`
- Go to Control Panel ... System ... Environment
- Append the following to the PATH variable (note the period at the end):
  `c:\jdk1.1.7\bin;.`
- Append the following to the CLASSPATH variable (note the period at the end) :
  `c:\jdk1.1.7\lib\classes.zip;c:\jdk1.1.7\lib\jcchart300.jar;.`
- Restart Windows for the changes to take effect.
- To run the program, get a DOS prompt and type:

  ```
  cd c:\Plotter
  javac Plotter.java
  java Plotter
  ```

4.2.2 Windows 95 and Windows 98

- Copy the Plotter folder to `c:\Plotter`
- Install JDK version 1.1.7 to `c:\jdk1.1.7` by following the installation instructions. Get JDK at `www.java.sun.com`.
- Copy `jcchart300.jar` to `c:\jdk1.1.7\lib`
- Add the following lines to `c:\autoexec.bat`

```
PATH c:\jdk1.1.7\bin;.
set
CLASSPATH=c:\jdk1.1.7\lib\classes.zip;c:\jdk1.1.7\lib\jcchart300.jar;.
```

[Note: if a PATH statement already exists, append `c:\jdk1.1.7\bin;.` to the end]

- Restart Windows for the changes to take effect.

- To run the program, get a DOS prompt and type:

```
cd c:\Plotter
javac Plotter.java
java Plotter
```

4.2.3 UNIX

- Copy the Plotter folder to `/home/joeuser/Plotter` (for example)

- Install JDK version 1.1.7 to by following the JDK installation instructions. Get JDK at `www.java.sun.com`

- Copy jcchart300.jar to `/home/joeuser/Plotter`

- Add the following line to the user *.login* file (note the period at the end):

```
setenv CLASSPATH /bin:/home/joeuser/Plotter/jcchart.jar:.
```

- Logout and login for the changes to take effect.

- To run the program type:

```
cd /home/joeuser/Plotter
javac Plotter.java
java Plotter
```

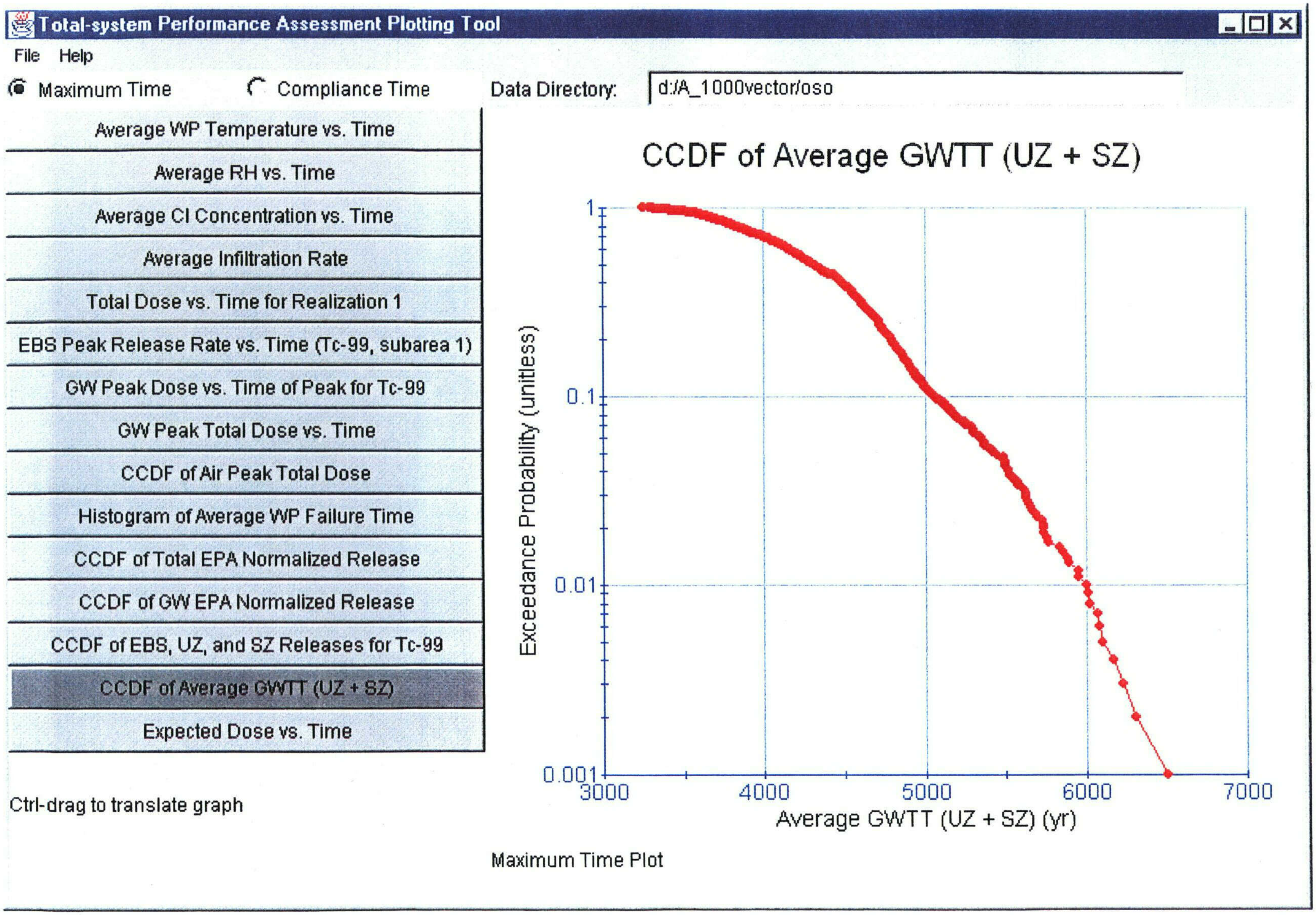
5 DISPLAYING THE PLOTS

The Java plotter can be executed and the application window can be invoked by executing the last step in the installation instructions in sections 4.2.1, 4.2.2, or 4.2.3. An example of the Java application window is shown in figure 1. If the shortcut icon is created and is located on the desktop, the application window can also be invoked by simply clicking on the plotter icon.The user must correctly specify the path to the directory from which data are to be plotted. This path is specified in the text field labeled "Data Directory" located near the top of the application window. The directory containing the *.plt* files may reside on any computer as long as it is accessible from the computer on which the Java processor is executed. On a Windows machine, the path might be something like:

```
c:\plotter\data\
```

On a UNIX machine, the path might be something like:

```
/home/users/joeuser/plotter/data/
```



● Maximum Time ○ Compliance Time

Data Directory: d:/A_1000vector/oso

Ctrl-drag to translate graph

Maximum Time Plot

Figure 1. The application window showing the result of the user selecting a maximum time plot of "CCDF of Average GWTT (UZ+SZ)" using a data file from the `d:\A_1000vector/oso` directory

54/58

Note that for a Windows machine, the separator character is "\" while for a UNIX machine, it is "/". The application will store the data path from the previous session in the file *plotter.ini*. The next time the application is executed, this data path automatically appears in the path field.

The user can select the time period of interest by clicking a button at the upper left corner of the application window and, for that time period, can select one of fifteen plots by clicking the corresponding button. These plots are briefly described in the following section.

6 DESCRIPTION OF PLOTS

This section provides a brief description of the plots, identifies the TPA output files from which data are read by the *fort_process.f* code, and discusses the processing of data for generating the *.plt* files. These descriptions can also be accessed from the Java postprocessor "Help" menu for both the maximum and compliance time periods. Descriptions of various modules and parameters can be found in Mohanty and McCartin (1998). None of the plots other than "Expected Dose vs. Time" plot have been weighted by the probability of occurrence of the disruptive event while computing release or dose. The average values are computed for the entire repository based on equal weighting of subareas and, thus, are independent of subarea sizes.

6.1 AVERAGE WASTE PACKAGE TEMPERATURE VS. TIME PLOT

The plot of "Average WP Temperature vs. Time" is a time history of the waste package (WP) temperature values from the NFENV module that are reported in the *nearfld.res* file at every tenth TPA time step. The WP temperature is averaged over all subareas with equal weights assigned to each subarea.

6.2 AVERAGE RELATIVE HUMIDITY VS. TIME PLOT

The plot of "Average RH vs. Time" is a time history of the relative humidity (RH) values from the NFENV module that are reported in the *nearfld.res* file at every tenth TPA time step. The RH is averaged over all subareas with equal weights assigned to each subarea.

6.3 AVERAGE CHLORIDE CONCENTRATION VS. TIME PLOT

The plot of "Average Cl Concentration vs. Time" is a time history of the chloride concentration values from the NFENV module that are reported in the *nearfld.res* file at every tenth TPA time step. The chloride concentration is averaged over all subareas with equal weights assigned to each subarea.

6.4 AVERAGE INFILTRATION RATE VS. TIME PLOT

The plot of "Average Infiltration Rate" is a time history of three infiltration rates: (i) the average infiltration rate from UZFLOW, (ii) the infiltration rate after reflux from the NFENV module, and (iii) the infiltration rate after diversion (using the F_{mult} and F_{ow} parameters). These values are reported in the *infilper.res* file at every tenth TPA time step. The infiltration rates are averaged over all subareas with equal weights assigned to each subarea.

6.5 TOTAL DOSE VS. TIME PLOT

The plot of "Total Dose vs. Time for Realization 1" is the time history reported in *totdose.res* of the total dose to the receptor group from all groundwater and ground surface radionuclides for realization 1. The total dose is the sum of the individual radionuclide doses calculated in DCAGW and DCAGS at each time step in the time period. These values are reported in the *totdose.res* file and provide total dose over the time period of interest at each TPA time step.

6.6 EBS PEAK RELEASE RATE VS. TIME PLOT

The plot of "EBS Peak Release Rate vs. Time (Tc-99, subarea 1)" is a scatter plot of the peak release rate of a radionuclide (Tc-99) from the engineered barrier system (EBS) and the corresponding time of the peak release rate for all realizations. These values are computed in EBSREL and are reported in the *pkreltim.res* file over the time period at each TPA time step.

6.7 GROUNDWATER PEAK DOSE VS. TIME OF PEAK FOR Tc-99 PLOT

The plot of "GW Peak Dose vs. Time of Peak for Tc-99" is a scatter plot of the peak groundwater dose of a radionuclide (Tc-99) to a receptor group and the corresponding time of the peak groundwater dose for all realizations. These values are computed in DCAGW and are reported in the *npkdoset.res* file over the time period at each TPA time step.

6.8 GROUNDWATER PEAK TOTAL DOSE VS. TIME PLOT

The plot of "GW Peak Total Dose vs. Time" is a scatter plot of the peak total groundwater dose to a receptor group and the corresponding time of the peak total groundwater dose for all realizations. The peak total groundwater dose is the maximum of the total groundwater dose which is the sum of individual radionuclide doses calculated in DCAGW. The peak total groundwater dose and the corresponding time of the peak dose over the time period are reported in the *gwpkdos.res* file.

6.9 CCDF OF AIR PEAK TOTAL DOSE PLOT

The plot of "CCDF of Air Peak Total Dose" is a CCDF of the peak total ground surface dose to a receptor group from an extrusive volcanic event. The peak total ground surface dose is the maximum of the total ground surface dose which is the sum of individual radionuclide doses calculated in DCAGS. The peak total ground surface dose over the time period is reported in the *airpkdos.res* file.

6.10 HISTOGRAM OF AVERAGE WP FAILURE TIME PLOT

The "Histogram of Average WP Failure Time" plot presents a histogram of the average WP failure time from corrosion and disruptive events (faulting, intrusive volcanic and seismic events). The WP failure times for all realizations are binned into different time intervals to generate data used to construct the histogram. The frequency in the histogram plot represents only the fraction of realizations for which the average WP failure time (not the number of failed WPs) is within a specified time interval. The average WP failure time for a realization is computed by weighting the time of corrosion and disruptive events reported in the *wpsfail.res* file with the corresponding number of failed WPs.

6.11 CCDF OF TOTAL EPA NORMALIZED RELEASE PLOT

The plot of "CCDF of Total EPA Normalized Release" is a CCDF of the sum of the groundwater and ground surface EPA normalized releases for all realizations. The EPA normalized release is computed using (i) SZFT results of the total amount of a radionuclide released from the saturated zone at the receptor location through well pumping over the time period for the groundwater and (ii) VOLCANO results of the total amount of a radionuclide released from the extrusive volcanic event for the ground surface. These releases are normalized using the EPA release limit of the radionuclide (U.S. Code of Federal Regulations, 1987). The total EPA normalized release for a realization is computed by summing the groundwater and ground surface releases for each radionuclide over all radionuclides without weighting the release with the probability of occurrence of the disruptive event. The total EPA normalized release over the time period is reported in the *relccdf.res* file.

6.12 CCDF OF GW EPA NORMALIZED RELEASE PLOT

The plot of "CCDF of GW EPA Normalized Release" is a CCDF of the groundwater EPA normalized releases for all realizations. The groundwater EPA normalized releases are computed using SZFT results of the total amount of a radionuclide released from the saturated zone at the receptor location through well pumping over the time period of interest. The release is normalized using the EPA release limit of the radionuclide (U.S. Code of Federal Regulations, 1987). The total EPA normalized release in the realization is computed by summing the groundwater releases for each radionuclide over all radionuclides. The total EPA normalized release over the time period of interest is reported in the *gwccdf.res* file.

6.13 CCDF OF EBS, UZ, SZ RELEASES FOR Tc-99 PLOT

The plot of "CCDF of EBS, UZ, and SZ Releases for Tc-99" consists of three CCDFs of the total release from the EBS, the unsaturated zone (UZ) and the saturated zone (SZ), over the time period of interest for one radionuclide (Tc-99). The three CCDFs are constructed from the total EBS, SZ, and UZ releases of the radionuclide for all realizations that are computed from EBSREL, SZFT, and UZFT results, respectively. The EBS, SZ, and UZ releases during the time period of interest are reported in the *cumrel.res* file.

6.14 CCDF OF AVERAGE GWTT (UZ + SZ) PLOT

The plot of "CCDF of Average GWTT (UZ + SZ)" is a CCDF of the sum of the average groundwater travel times (GWTT) for the unsaturated zone (UZ) and saturated zone (SZ). These travel times represent the average GWTT for all subareas with equal weighting for each subarea. In each realization, the UZ GWTT is computed in UZFT and the SZ GWTT is calculated in SZFT for each subarea. The subarea averaged UZ and SZ GWTT values are reported in the *gwttuzsz.res* file.

6.15 EXPECTED DOSE VS. TIME PLOT

The plot of "Expected Dose vs. Time" presents the expected total dose as a function of time. The expected dose is the average of the total dose from all realizations at each time step. The expected total dose curve is not weighted by the scenario probability. The scenarios are specified in the tpa.inp file and are designated as *oso*,*fso*, and *osv* which correspond to the basecase with seismicity, with faulting and seismicity, and with seismicity and volcanism, respectively. The time history of the expected doses is reported in the *rgwsa.tpa* file.

7 PRINTING THE PLOTS

To produce hardcopy output of the current plot, the user may select "Print Current Plot..." from the File menu. The printed size of the plot is determined by the size of the plot on the computer screen. The plot can be resized on the computer screen before printing if a different plot size is needed. On a Windows platform, one can save the plot for later use by selecting "Print to File" in the print dialog box, and the plot will be saved as a postscript file.

8 USER SUPPORT

For technical assistance, users may contact

Sitakanta Mohanty
Center for Nuclear Waste Regulatory Analyses
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510
(210) 522-5185
smohanty@swri.org

Hollis A. Thomas
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510
(210) 522-4958
hthomas@swri.org

9 REFERENCES

Mohanty, S., and T.J. McCartin. 1998. *Total-system Performance Assessment (TPA) Version 3.2 Code: Module Descriptions and User's Guide*. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

U.S. Code of Federal Regulations. 1987. *Environmental Standards for the Management and Disposal of Spent Nuclear Fuel, High-Level and Transuranic Radioactive Wastes.*Title 40—Protection of the Environment, Chapter 1—Environmental Protection Agency, Part 191. Washington, DC: U.S. Government Printing Office.

58/58

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

To: TPA 3.2 AP
Software Folder

December 3, 1998
Contract No. NRC-02-97-009
Account No. 20-1402-762

U.S. Nuclear Regulatory Commission
ATTN: Mr. James Firth
Office of Nuclear Materials Safety and Safeguards
Division of Waste Management
Performance Assessment and HLW Integration Branch
Mail Stop 7C-18
Washington, DC 20555

Subject: Transmittal of the TPA Version 3.2 Code Post-Processor and PVM Version of the TPA Version 3.2 Code

Dear Mr. Firth:

The purpose of this letter is to transmit TPA Version 3.2 Code Post-Processor—AI 1402-762-808 and PVM Version of the TPA Version 3.2 Code—AI 20-1402-762-807. Delivery of the PC Version of the TPA Version 3.2 Code—AI 20-1402-762-809 will be postponed until next week to allow more thorough review of the installation and execution guide.

Attached herewith are diskettes containing copies of the Java source code for the post-processor and a tape containing FORTRAN source code and executable code for the PVM Version of the TPA Version 3.2 code. As we agreed, the PARJOB routines used in the PVM Version, which were developed and copyrighted by Southwest Research Institute will be supplied only in executable form. All FORTRAN source code developed specifically for the PVM Version is, of course, supplied as source code. If the NRC CRADAL system is reconfigured to include computers that are not binary compatible with Sun SPARC processors, we will supply recompiled PARJOB routines. Moreover, if NRC receives requests from DOE for the PVM Version, we will provide DOE with executable code for their specific computer system.

Also attached are installation and execution guides for both software products. If you have any questions regarding the installation and use of the software or the technical content of the guides please contact Dr. Sitakanta Mohanty at (210) 522-5185.

Sincerely yours,

Gordon Wittmeyer
Gordon W. Wittmeyer, Ph.D.
Manager, Performance Assessment

GWW/cg

cc:
J. Linehan
D. DeMarco
B. Stiltenpole
B. Meehan
J. Greeves
J. Holonich
M. Bell
K. McConnell
T. McCartin
R. Codell
W. Patrick
CNWRA Directors
CNWRA Element Managers
S. Mohanty
R. Janetzke

Washington Office • Twinbrook Metro Plaza, #210 • 12300 Twinbrook Parkway • Rockville, Maryland 20852-1606

1A/6

SOFTWARE DEVELOPMENT PLAN

1/6

CE[illegible]R FOR NUCLEAR WASTE REGULA[illegible]Y ANALYSES
DOCUM[illegible]T REVIEW REQUEST AND TRANSMITTAL C[illegible]TROL (REF. QAP-002)

I. DOCUMENT INFORMATION

a. TITLE: (1) Software Requirements Description - Supermodsign Post-Processor for TPA Version 3.2
(2) Software Development Plan - Supermodsign Post-Processor for TPA Version 3.2

b. DOCUMENT TYPE

☐ Technical Report ☐ AP ☐ RPD ☐ Paper/Presentation * ☐ Project/Test Plan
☒ Guidance Document ☐ TOP ☐ CQAM/QAP ☐ OPs/Work Plan ☐ Proposal

* Conference/Journal Title:

Special Markings ***(such as "Predecisional" or "Proprietary")*** **Yes** ____ **No** ____

c. PROJECT INFORMATION

Project No. 20-1402-761 Milestone No. ____ Subject Code 707.2

CNWRA DOCUMENT NO. Yes ____ No ____ Assigned No. CNWRA 97 —

d. SCHEDULE Today's Date April 2, 1999 Scheduled Transmittal Date 4/14/99

II. RESPONSIBILITIES (Fill in names on each blank line in this section.)

Author(s) K. Poor Element Manager Gordon wittmeyer Assigned Secretary Cathy Garcia

III. REVIEW (See QAP-002 table 1 for applicable review types.)
Review Types & Reviewers Determined by Element Manager

Gordon Wittmeyer (EM Signature) 4/2/99 (Date)

		Req'd Date	Initials	Completed
☒	TECHNICAL (Attach CNWRA form QAP-12.) Reviewer(s): James Weldy	4/6/99	JRW	4/5/99
☐	PEER (Attach CNWRA form QAP-13.) Reviewer(s):			
☐	EDITORIAL Reviewer:			
☐	CONCURRENCE Reviewer:			
☐	QA Reviewer:			
☒	PROGRAMMATIC Reviewer: Wesley Patrick	4/9/99	WCP	4/7/99
☒	FORMAT Reviewer: Bonnie Caudle	4/13/99	BC	4/7/99
	Verification of Compliance with QAP-002		JCJ for BEm	4/7/99

IV. TRANSMITTAL
TO: ____ FROM: ____

COPIES TO: (Add/delete names as required using current information in "Guidelines for Minimum Distribution of CNWRA Correspondence.")

Distribution (listed below)

2/2

CENTER FOR NUCLEAR WASTE REGULATORY ANALYSES
INSTRUCTIONS TO TECHNICAL REVIEWERS
Technical Review Items to Verify

TO: James Weldy

SUBJECT: Review of
(1) Software Requirements Description - Supermodsign Post-Processor for TPA Version 3.2
(2) Software Development Plan - Supermodsign Post-Processor for TPA Version 3.2

Please perform a Technical Review of the subject document in accordance with CNWRA QAP-002, verifying the specific items identified below. Technical comments shall be documented on the attached Comment Resolution Record and presented to the author for resolution. Initial blanks on right side of page to show completion of assigned review.

Required review completion date: April 6, 1999

ASSIGNED **ACCOMPLISHED**

TECHNICAL CORRECTNESS

X	Assumptions are reasonable and clearly stated.	JRW
	Appropriate techniques are used.*	
	Computations are correct, calculations are documented and verified in accordance with QAP-014 (document this review by a statement on the TOP-3 form).	
	Existing data are qualified (or exempted) in accordance with QAP-015.	
	Conclusions are properly supported by correctly interpreted data.*	

* *Novel or beyond state-of-the-art techniques or significant uncertainties in data and interpretations warrant application of the Peer Review.*

READABILITY

X	Document is written for the intended audience, with correct grammar and syntax.	JRW
X	Illustrations and tables clearly present basic information and emphasize relationships.	JRW

CONTENT AND FORMAT

X	Title reflects the objectives of the document.	JRW
	Abstract states purpose, describes study, and summarizes the pertinent results and conclusions.	
X	Introduction states the objectives and scope of the work and presents background information.	JRW
X	Body of the manuscript is logically organized and presents the basic information.	JRW
X	Conclusions and results summarize the principal findings and answer each of the objectives of the work.	JRW
X	References are cited in the text and in the references section.	JRW
	Costs and financial tables are included and agree with text.	

ELEMENT MANAGER	DATE	COGNIZANT DIRECTOR	DATE
Gordon Wittmeyer	4/2/99	for BS	4/2/99

NWRA FORM QAP-12-4 (Rev.6/98)

CNWRA REPORT REVIEW / COMMENT RESOLUTION RECORD

PAGE 1 OF 1 PAGES

PROJECT NUMBER 20-1402-761	DOCUMENT DATE April 14, 1999	DOCUMENT NUMBER 20-1402-761-900

TITLE: (1) Software Requirements Description - Supermodsign Post-Processor for TPA Version 3.2
(2) Software Development Plan - Supermodsign Post-Processor for TPA Version 3.2

The comments shown below address questions and concerns of a technical and/or programmatic nature which arose in this review. Because of possible implications, they require action and response.	RESPONSE: (Write "accept" and note briefly how comment was incorporated, or give justification if rejected.)
1. Editorial Suggestions, as marked.	
2 These documents appear to meet TOP-018 requirements - no technical issues identified	

REVIEWER SIGNATURE: James L. Weldy DATE: 4/15/99

RESPONDER SIGNATURE: DATE:

Response accepted by: James L. Weldy Signature Date 4/15/99

If resolution cannot be achieved, the matter shall be elevated to the next level of authority.

Distribution: This completed form shall be maintained in a record file.

CNWRA Form TOP-3 (Rev. 6/90)

CNWRA REPORT REVIEW / COMMENT RESOLUTION RECORD

PAGE 1 OF 3 PAGES

PROJECT NUMBER 20-1402-761

DOCUMENT DATE April 14, 1999

DOCUMENT NUMBER 20-1402-761-900

TITLE: (1) Software Requirements Description - Supermodsign Post-Processor for TPA Version 3.2
(2) Software Development Plan - Supermodsign Post-Processor for TPA Version 3.2

The comments shown below address questions and concerns of a technical and/or programmatic nature which arose in this review. Because of possible implications, they require action and response.	RESPONSE: (Write "accept" and note briefly how comment was incorporated, or give justification if rejected.)
1. See minor comments in text.	① Thank You
2, SRD, p. 1, para. 1. The discussion presents an argument that justifies the planned approach without defining either event or parameter trees. Suggest inserting 1-2 sentences before concluding "Since event trees . . . ."	② Insert has been added.
3. NOTE: Jarzemba and Sagar was assumed to be separately reviewed and ~~was~~ is not addressed in these comments.	③ This is a valid assumption - Jarzemba and Sagar (1999) has been reviewed as a part of a PA deliverable.
4, SRD, p. 1, last para. Suggest clarifying this (e.g., "generated" is ambiguous).	④ We are using "caused" as suggested

REVIEWER SIGNATURE:

DATE: 4/6/99

Response accepted by:

Signature

Date 4/7/99

RESPONDER SIGNATURE: Mark Jarzemba

DATE: 4-6-99

If resolution cannot be achieved, the matter shall be elevated to the next level of authority.

Distribution: This completed form shall be maintained in a record file.

CNWRA Form TOP-3 (Rev. 6/90)

CNWRA REPORT REVIEW / COMMENT RESOLUTION RECORD

PAGE 2 OF 3 PAGES

PROJECT NUMBER	DOCUMENT DATE	DOCUMENT NUMBER
20-1402-761	April 14, 1999	20-1402-761-900

TITLE: (1) Software Requirements Description - Supermodsign Post-Processor for TPA Version 3.2
(2) Software Development Plan - Supermodsign Post-Processor for TPA Version 3.2

The comments shown below address questions and concerns of a technical and/or programmatic nature which arose in this review. Because of possible implications, they require action and response.	RESPONSE: (Write "accept" and note briefly how comment was incorporated, or give justification if rejected.)
5. SRD, sec. 3.2. Will this code be developed only for NT or UNIX, or for both? Seems like the need is for both versions. Clarify.	⑤ No, it will be platform independent (standard FORTRAN90). Words have been added to explain this.
6. SDP, sec. 1. Further to Com. #5, the discussion here indicates both systems will be addressed.	⑥ Similar language to that above has been added.
7. SDP, sec. 3.4. Need to add brief discussion on WBS (charge number) and introduce the tables.	⑦ What you wrote looks good and is added
8. SDP, sec. 4.2. Consistent with Institute EEO requirements, we don't inquire as to the lifestyle of Codes we employ. ☺	⑧ OOPS. Amazing what spell-checkers will do sometimes.

REVIEWER SIGNATURE:	DATE:	RESPONDER SIGNATURE:	DATE:
	4/6/99	Mark Jarzem	4-6-99

Response accepted by: Signature Date 4/7/99

If resolution cannot be achieved, the matter shall be elevated to the next level of authority.

Distribution: This completed form shall be maintained in a record file.

CNWRA Form TOP-3 (Rev. 6/90)

CNWRA REPORT REVIEW / COMMENT RESOLUTION RECORD

PAGE 3 OF 3 PAGES

PROJECT NUMBER	DOCUMENT DATE	DOCUMENT NUMBER
20-1402-761	April 14, 1999	20-1402-761-900

TITLE: (1) Software Requirements Description - Supermodsign Post-Processor for TPA Version 3.2
(2) Software Development Plan - Supermodsign Post-Processor for TPA Version 3.2

The comments shown below address questions and concerns of a technical and/or programmatic nature which arose in this review. Because of possible implications, they require action and response.	RESPONSE: (Write "accept" and note briefly how comment was incorporated, or give justification if rejected.)
9. SDP, sec. 4.2. Add a sentence to introduce what follows to the reader.	⑨ Introductory sentence has been added.
10. SDP, sec. 4.4. Need to clarify where/how this will be documented (e.g., scientific notebook).	⑩ Have added a statement that documentation of testing and proposed changes will be done using accepted CNWRA QA methods (like the scientific notebook).
11. SDP, sec. 4.4. All NRC feedback should be to the CNWRA, regardless of who does the actual programming.	⑪ Change accepted
12. SDP, sec. 5.3. Suggest clarifying the limited period during which normal check-in/check-out is suspended.	⑫ Suggested addition accepted

REVIEWER SIGNATURE: DATE: 4/6/99

RESPONDER SIGNATURE: Mark D. Jarzemba DATE: 4-6-99

Response accepted by: Signature Date 4/7/99

If resolution cannot be achieved, the matter shall be elevated to the next level of authority.

Distribution: This completed form shall be maintained in a record file.

CNWRA Form TOP-3 (Rev. 6/90)

1/5

SOFTWARE DEVELOPMENT PLAN - SUPERMODSIGN POST-PROCESSOR FOR TPA VERSION 3.2

by

Kevin Poor

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

Reviewed by:

4/8/99

James Weldy

Approved by:

Gordon Wittmeyer 4/8/99

Gordon Wittmeyer, Element Manager, Performance Assessment

SOFTWARE DEVELOPMENT PLAN - SUPERMODSIGN POST-PROCESSOR FOR TPA VERSION 3.2

1 SCOPE

Supermodsign is a post-processor tool for developing parameter trees from Total-system Performance Assessment (TPA) Version 3.2 code results. Since parameter trees are similar to event trees, which are commonly used to present risk assessment results for other complicated systems (e.g., nuclear reactors), it is expected that their application to repository performance assessment (PA) would help make the results of complex PA models more transparent. The application will be written in standard FORTRAN90 and hence it will have no platform- or system-specific requirements. A brief users guide will accompany the software.

2 BASELINE ITEMS

- Parameter tree post processor for TPA version 3.2 output
- Code to read the data files: reads and formats data from a TPA run
- Test Data: sample TPA run data used to test the program
- Users Guide: describes computational algorithms, system requirements, and code usage procedures

3 PROJECT MANAGEMENT

3.1 PROJECT SCHEDULE

Work will begin in late-March 1999 and continue through the end of June 1999.

3.2 STAFFING

One CNWRA staff member working part time with limited contractor support have been assigned for the duration of the project.

3.3 RISK MANAGEMENT

The current scope of this project is to develop a stand-alone code as described in Jarzemba and Sagar (1999). Additional development may be required in the future to incorporate the Supermodsign code into existing TPA post processors. Upcoming changes in CNWRA staff availability may affect code development. To allow completion of code development and testing on schedule, additional PA element subcontractor support is being considered.

3.4 WORK BREAKDOWN STRUCTURE

All work will be completed under the TPA code development component of the work breakdown structure (i.e., 20-1402-762). Required labor is estimated in the following table.

Task	Estimated Labor Hours
Generate preliminary system requirements	10
Develop code to read TPA output files	20
Test and refine data read capability	10
Develop code to generate parameter trees and analysis results for user defined TPA input parameters	100
Develop code to serially evaluate (i.e., in a stepwise manner) all stochastic TPA input parameters to develop parameter tree based on an importance factor	80
Test and refine parameter tree development and analysis capability	80
Test cross-platform portability	10
Produce User's Guide	40
Modify code based on customer feedback	40
Prepare project management administrative paperwork	10
Total	400

4 DEVELOPMENT PROCEDURES

4.1 HARDWARE AND SOFTWARE RESOURCES

The software will be developed on a Pentium-based PC running Windows NT. It will be ported to and tested on a Pentium-based PC running Windows 95/98 and a Sun work station running Solaris. These hardware resources are already available in-house and do not need to be purchased.

4.2 SOFTWARE DEVELOPMENT LIFECYCLE

The following describes events in the development of the herein described software:

Analysis—determine input data format, formulate requirements for interface, determine output requirements and format.

Phase I product development—develop code input and output capability, develop parameter tree and analysis algorithms for user-defined TPA stochastic input parameters.

4/5

Phase I product testing—test phase I product parameter tree development and analysis algorithms with commercially available spreadsheet sorting routines. Additionally, test program data input and output capabilities.

Phase II product development—develop Supermodsign code to allow multiple level parameter tree development and identification of those input parameters that most affect output response by sequential (i.e., stepwise) analysis of all TPA input parameters.

Phase II product testing—test phase II product parameter tree development and analysis algorithms with commercially available spreadsheet sorting routines.

Iteration release—developers release a version of the software to users.

Testing and user feedback—users provide developers feedback on "look and feel" and functionality of product. Developer uses this information to develop final version of software.

Final delivery—developers provide final version of software to users.

4.3 CODING

The Supermodsign TPA post processor will be written in standard Fortran 90.

4.4 ACCEPTANCE TESTING AND ANALYSIS

SwRI or subcontractor personnel will perform preliminary acceptance testing on the Supermodsign software. Testers will document results of testing and proposed changes using accepted CNWRA Quality Assurance methods (e.g., scientific notebook). The code developers will use this information to make revisions to the program. After preliminary acceptance testing, the end user (the U.S. Nuclear Regulatory Commission) will use the software and provide comments and change requests to the CNWRA, so that the code can be further modified.

5 CONFIGURATION MANAGEMENT PLAN

5.1 TOOLS

Due to the relatively small size of the program, no special configuration management tools are required for this project.

5.2 CONFIGURATION IDENTIFICATION

The code to develop and analyze parameter trees, the code test data, and the User's Guide will be placed under configuration control after customer acceptance. In order to place a particular release version under configuration control, the developers will create a folder named Supermodsign*mmdd*, where *mmdd* is the date the folder was created. This folder will be archived.

5/5

5.3 CONFIGURATION PROCEDURES

Due to the small size of the Supermodsign program and development staff, no check-in/check-out procedures are required during the code development and testing phase. Release versions of the Supermodsign software will be cleared through and approved by CNWRA personnel. No official documentation such as an SCR is required for changes to the software during preliminary acceptance testing. Once the code is baselined and ready for delivery to the end user, the end user may request changes to the software using an SCR.

6 REFERENCES

Jarzemba, M.S., and B. Sagar. 1999. A Feasibility Study for a TPA Version 3.2 Event-Tree Post Processor. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.