SMLEXC: A SMALL EXECUTIVE CODE FOR SUBSTANTIALLY COMPLETE CONTAINMENT EXAMPLE ANALYSIS OF A REFERENCE CONTAINER

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-93-005

Prepared by

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

September 1994

SMLEXC: A SMALL EXECUTIVE CODE FOR SUBSTANTIALLY COMPLETE CONTAINMENT EXAMPLE ANALYSIS OF A REFERENCE CONTAINER

Prepared for

Nuclear Regulatory Commission
Contract NRC-02-93-005

Prepared by

Ronald W. Janetzke
Tony Y. Torng
Peter C. Lichtner

Center for Nuclear Waste Regulatory Analyses
San Antonio, Texas

September 1994

PREVIOUS REPORTS IN SERIES

Number	Name	Date Issued
CNWRA 94-003	"Substantially Complete Containment"—Example Analysis of a Reference Container	January 1994
Letter Report	SCCEX: A Computer Code for Substantially Complete Containment Example Analysis of a Reference Container	July 1994

ABSTRACT

The SMaLl EXeCutive (SMLEXC) computer code is designed primarily to provide a computational framework using a probabilistic approach to construct cumulative distribution functions (CDFs) and probability density functions (PDFs) of waste package failures in a repository. The CDFs and PDFs are used to evaluate containment by waste packages and to identify those areas requiring more detailed analysis. The SMLEXC code contains five basic codes: (i) an executive code to control the input and output of other codes; (ii) a probabilistic analysis code (Latin Hypercube Sampling) to sample data from statistical distributions; (iii) a preprocessor code to prepare input for the Substantially Complete Containment EXample (SCCEX) code; (iv) a deterministic analysis code (SCCEX) to compute time-to-wetting and time-to-failure for each calculation cell; and (v) a postprocessor code to construct CDFs and PDFs from calculated wetting and failure data. In this user's manual, the various codes in the SMLEXC code, code structure and capabilities, input and output files for a baseline case, and configuration control status are outlined. This code was used to solve an example problem for evaluating the "substantially complete containment" aspects of the Code of Federal Regulations, Title 10, Part 60 (10 CFR Part 60).

CONTENTS

CONTENTS (cont'd)

FIGURES

TABLES

ACKNOWLEDGMENTS

This report was prepared to document work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Nuclear Regulatory Commission (NRC) under Contract No. NRC-02-93-005. The activities reported here were performed on behalf of the NRC Office of Nuclear Material Safety and Safeguards (NMSS), Division of Waste Management (DWM). The report is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC.

The authors gratefully acknowledge the technical review of Dr. Ross Bagtzoglou and the programmatic review of Dr. Budhi Sagar. Appreciation is due to Bonnie L. Garcia and Alana Woods for their assistance in the preparation of this report.

QUALITY OF DATA

DATA: Sources of data are referenced in each chapter. The respective sources of these data should be consulted for determining their levels of quality assurance. The SMLEXC code is controlled at the CNWRA according to the provisions of Technical Operating Procedure (TOP)-018.

1 INTRODUCTION

The objective of the SMaLl EXeCutive (SMLEXC) computer code is to provide a computational framework using a probabilistic approach to construct cumulative distribution functions (CDFs) and probability density functions (PDFs) of waste package failures in a repository. The CDFs and PDFs are used to evaluate containment by waste packages and to identify those areas requiring more detailed analysis. However, the SMLEXC code is not intended to demonstrate the suitability of any given waste package material or design in terms of performance requirements. This code was used to solve an example problem for evaluating the "substantially complete containment" aspects of the Code of Federal Regulations, Title 10, Part 60 (10 CFR Part 60) (1990).

The assessment of containment of high-level waste (HLW) in a waste package involves the use of several interconnected computational models. To conduct such an assessment, an appropriate deterministic analysis model was developed to compute time-to-wetting and time-to-failure of waste packages. This analysis model takes into account the repository thermal field; near-field geochemical environment; corrosion of container materials, including localized corrosion, stress corrosion cracking, and mechanical failures; and release of radionuclides.

In the evaluation of containment, it is recognized that the waste packages could be exposed to varying environmental conditions. Also, there are uncertainties in the parameters used in various models (e.g., the age of fuel, environmental factors, corrosion constants, and rock properties, etc.). Deterministic analyses are unable to account for these uncertainties. In the SMLEXC code, these uncertainties in the repository parameters are modeled as a number of correlated random variables with selected distributions. In the end, the CDFs and PDFs of waste package failures are computed. The CDFs and PDFs are then interpreted to assess whether "substantially complete containment" requirements of 10 CFR Part 60 are satisfied.

The SMLEXC software uses a collection of several independent codes that are linked together to function as a unit. The five basic parts of the SMLEXC code are: (i) an executive code to control the input and output of other codes; (ii) a probabilistic analysis method, namely the Latin Hypercube Sampling (LHS), for sampling from statistical distributions; (iii) a preprocessor for preparing the Substantially Complete Containment EXample (SCCEX) code input; (iv) the SCCEX code for performing deterministic analysis by computing time-to-wetting and time-to-failure given a set of sampled data; and (v) a postprocessor to construct CDFs and PDFs from calculated wetting and failure data.

A brief overview of the codes used in the SMLEXC Version 1.0 software is provided in Section 2. Section 3 discusses the capability and structure of the SMLEXC code, the mechanics of data transfer between various codes, and various features required by the SMLEXC code. Section 4 contains the input file descriptions for the executive code and other codes. The output file descriptions are given in Section 5. Section 6 contains a brief summary and conclusion.

2 DESCRIPTION OF SMLEXC CODE

2.1 OVERVIEW

The SMLEXC software contains five independent codes that are linked together to achieve its intended objectives. These codes include: (i) an executive code; (ii) an LHS code; (iii) a preprocessor (SCCEXP) code; (iv) the SCCEX code; and (v) a postprocessor (POSTDD) code.

The executive code acts as the manager and assures that codes are executed in the desired sequence and that appropriate values of the common parameters are passed to the other codes. Automated features included in the SMLEXC system facilitate execution of a set of multiple realizations with an associated output.

The LHS code, which is capable of sampling correlated random variables, is used for sampling data for Monte Carlo analyses. This LHS code is called first by the executive code to develop a number of data sets (user-selected) based on the inputted random variables.

For each sampled data set, the executive code calls the SCCEXP code to map these sampled data into a SCCEX code input file. Once the executive code has this input file, the executive code then calls the SCCEX code to compute time-to-wetting and time-to-failure results. The above SCCEXP/SCCEX process is iterated in the executive code until all LHS sampled data sets are calculated. All time-to-wetting and time-to-failure results are stored in the file, postdd.dat.

The POSTDD code is then used to extract important information from the postdd.dat file to construct CDFs and PDFs to compare with the "substantially complete containment" aspects of 10 CFR Part 60.

An application of the SMLEXC code to an example problem can be found in the report entitled "Substantially Complete Containment"—Example Analysis of a Reference Container (Cragnolino et al., 1994). In the following subsections, a brief description of each code is provided.

2.2 THE EXECUTIVE CODE

The SMLEXC codes (LHS, SCCEXP, SCCEX, and POSTDD) that are linked together by the executive code are programs that can be executed by themselves (standalone) without the aid of the SMLEXC system. These programs are also referred to as subprocesses in the sense that they are spawned by a main computer process, the executive. The standard documentation prepared for these programs (or subprocesses) also applies to their use in the SMLEXC system, with minor modifications to input and output procedures.

The only calculations performed in the executive consist of sorting event times and plotting a CDF for each event. Times are sorted in a sequence from smallest to largest. Once the sorting is complete, cumulative probabilities are assigned to each time. Since the calculated times are statistically independent of each other, the estimate of the probability that the n^{th} time will be equalled or exceeded is $(N-n+1)/N$ where N is the total number of times in the sequence. For example, if 500 samples are used ($N=500$) and the 100^{th} time ($n=100$) in the sorted sequence is t_{100}, then

$p\{T > t_{100}\} = (500-100+1)/500 = 0.802$, where T is the random variable representing the actual time. The probability estimates improve as N increases. The best resolution that can be obtained in probability value is $1/N$. Normally, a convergence test should be performed to find the number of samples required such that the shape of the CDF is not altered significantly by the addition of more samples.

2.3 THE LHS CODE

The random nature of some of the parameters characterizing a realization is accommodated by using the Monte Carlo method. A set of parameter vectors, which is a user input data set transferred from the executive code, is created by sampling from the probability distributions of the random variables. The sampling method is implemented in the LHS code. The LHS code utilizes the LHS method (Iman and Shortencarrier, 1984) to create equally likely parameter vectors. Thus, if N vectors are included in a set, then each vector has a probability of occurrence of $1/N$. Simulations are performed with each vector in turn, resulting in the calculation of N realizations. It is then obvious that each realization is also equiprobable with a probability of $1/N$.

The LHS code is designed to sample from correlated variables as well as uncorrelated random variables. In short, the LHS method is based on dividing the probability space into N number of equal and discrete subspaces. In the case of a single variable, for example, the probability distribution is divided into N parts, where the area under each part is the same. A sample is then drawn randomly from each discrete part. Once the N samples are obtained, their order is randomized to obtain a vector made up of unordered components.

The theory for the correlated variables is complex, and the interested reader can refer to Iman and Shortencarrier (1984). The LHS program was obtained from Sandia National Laboratories (SNL). The output of the LHS is written to a file whose size depends upon the number of samples (or vectors) and the number of values in each sample. For instance, if there are n uncertain variables and N samples are required, then the output file will contain N vectors, each vector containing n values, or a total of $(N \times n)$ values. For further analysis, vectors are read one at a time from this file. Two aspects are to be noted: (i) all uncertain parameters are sampled at one time, and (ii) a single call to the LHS provides all of the samples for all of the vectors.

As discussed, the LHS code can be used to perform a probabilistic analysis; however, when the probability of failure becomes small, a large number of LHS samples is required to provide an accurate prediction. Other probabilistic schemes (e.g., probabilistic global/local approach) may have to be designed to assess small probability consequences.

2.4 THE SCCEXP CODE

The SCCEXP is an input filter or preprocessor for the SCCEX code. It generates the SCCEX input file from a combination of sources. One source is the default SCCEX input file. Another source is the LHS output file. The SCCEXP first reads the default SCCEX input file to set the default values for all of the parameters. These values are used for variables that are not sampled by the LHS code. Then, the LHS output file is read to select the appropriate values for all of the sampled variables in a

particular realization. Finally, the SCCEXP code generates an ASCII formatted input file with the correct values for the SCCEX code.

2.5 THE SCCEX CODE

Each invocation of the SCCEX code provides event times (time-to-wetting and time-to-failure) for each cell under consideration. The idealized repository is assumed to be a square area with an assumed one-eighth symmetry for the temperature field containing equally spaced waste packages. Based on this symmetry, the SCCEX code divides the repository into calculation cells. Depending on the total number of calculation cells used, each calculation cell could represent a large number of waste packages. A typical cell structure for the repository is shown in Figure 2-1. In this figure, the calculation cells on the diagonal occur four times in the repository, whereas, off-diagonal cells occur eight times. The number of times each cell occurs in the repository is called the weight factor, which is used to calculate the total number of cell (or waste packages) failures in the repository.

The SCCEX code performs the main core of the computation by considering the repository thermal field, near-field environment, corrosion of container materials (e.g., localized corrosion and stress corrosion cracking), as well as mechanical failure. The SCCEX code is designed to compute time-to-first-wetting and time-to-failure for each waste package within the repository. For simplicity, the failure of a waste package in this analysis is defined as the condition of a waste package in which localized corrosion has penetrated the container walls either by stress corrosion cracking or by mechanically dominated processes.

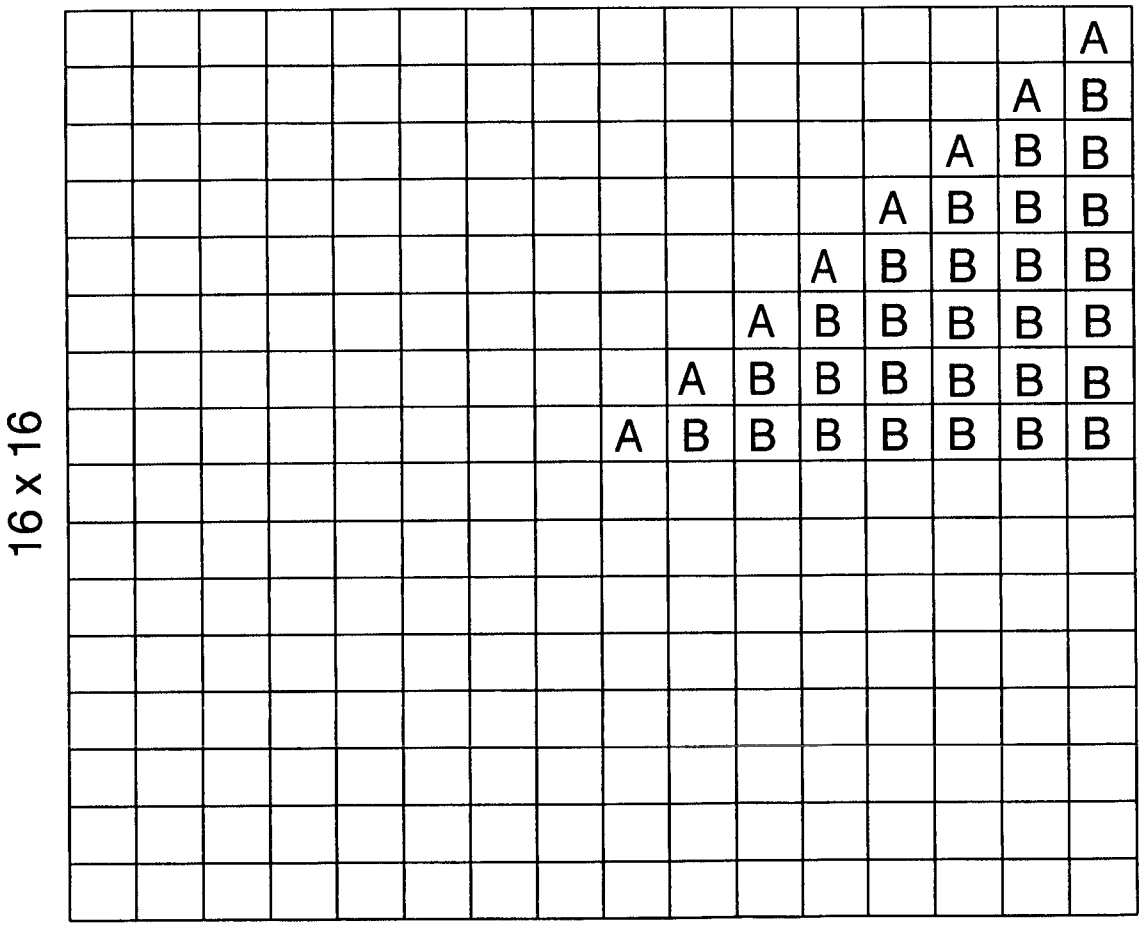
The SCCEX code is invoked by the SMLEXC system and uses the input file prepared by the SCCEXP code. The program completes its execution, producing the output as described in the SCCEX User's Guide (Torng et al., 1994). In addition, more detailed rationale, descriptions, and limitations of the various models used in the SCCEX code can be found in the Engineered Barrier System Performance Assessment Codes (EBSPAC) progress report (Sridhar et al., 1993) and the report entitled "Substantially Complete Containment"—Example Analysis of a Reference Container (Cragnolino et al., 1994).

2.6 THE POSTDD CODE

The process of SCCEXP/SCCEX is run until the all sampled data sets (from the LHS code) have been executed. Time-to-failure and time-to-wetting results for all calculation cells and vectors are processed by the executive code and stored in the data file postdd.dat.

Each time event datum (e.g., time-to-failure) is sorted by the executive code, and an empirical CDF is constructed according to each calculation cell's weight factor. This approach, as discussed in Section 2.2, produces the failure (or wetting) percentage; however, it neglects the location of the cell. Furthermore, because this approach is based on cells, each cell including a large number of waste packages, it may not provide sufficient detail as may be required by the Substantially Complete Containment (SCC) rule.

Alternatively, a postprocessor code POSTDD is used to analyze the data storage file postdd.dat. The POSTDD code is designed to extract specific time-to-failure data from the postdd.dat file for constructing either a CDF or a PDF. The time-to-wetting data are not used by the POSTDD code. The POSTDD Code determines: (i) the worst cell, (ii) CDF for time-to-failure for a user specified number



Note: A = 4 cells, B = 8 cells, modeled as 1/8 symmetry

Figure 2-1. Definition of repository cells and groups

of cell failures, and (iii) the PDF describing the number of cell failures for a user-specified time. These are discussed in more detail below.

(i) To keep track of the spatial location of each cell, the POSTDD code first calculates the time-to-failure statistics by collecting the time event data at each cell location from all vectors. The statistics are then compared to determine the ranking of each cell. Depending on the user-selected cell ranking, the time event data for that specific cell ranking is extracted to construct a CDF. With the critical cell location identified, if necessary, a more detailed analysis within this cell can be performed to acquire the waste package failure information. Note that a maximum of 10 cell rankings can be selected by the user.

(ii) To determine the time-to-failure for a user-specified number of cell failures, POSTDD code first sorts every time-to-failure event in each vector into ascending order. Depending on the user-selected number of cell failures, the time event data for that specific number of cell failures is collected from each vector to construct a CDF. The mean value of this CDF can also be translated to the average time-to-failure for the user-selected number of cell failures. Note that the user can specify a maximum of 10 number of cell failures.

(iii) To know the statistic of how many cell failures occur within a user-specified time (years), the POSTDD code first sorts every time-to-failure event in each vector into an ascending order. Depending on the user-selected time, the number of failure data for that specific time is collected from each vector to construct a PDF. This PDF can further be used to determine the probability that the proportion of cell (or waste package) failure during the specified time period. Note that the user can select a maximum of 10 time events.

To construct the CDFs and PDFs based on the extracted data, the POSTDD code uses a simple statistical strategy to analyze the extracted data. It is assumed that a total of N data have been collected. The first step of the POSTDD code is to sort the extracted data into ascending order. An empirical CDF is then constructed as $(i-0.5)/N$, where i represents the i^{th} data number. As for the PDF, an appropriate time interval needs to be determined first, based on the range of sorted data. The number of data, n_j, where j represents the j^{th} time interval, that falls within each interval is collected. An empirical PDF is then constructed as n_j/N.

Because data may be extracted at different times, different cells, or a different cumulative number of failures, it is not convenient to couple this code directly with the executive code. However, if the data of interest, such as the required design life, are known in advance, the POSTDD code can be easily integrated with the executive code. A detailed user instruction on how to input these data is summarized in Appendix A.

3 SMLEXC SOFTWARE DESCRIPTION

As discussed in Section 2, the SMLEXC software contains five individual codes. The executive code is used to control and interface with the rest of the codes (or subprocesses) which can be performed as an independent computer program with its own set of input and output files. In addition to the input/output (I/O) files, the executive code creates several additional files to transfer global data from one code to another, to record the log entry, the error messages, the subprocess execution, and to store data for plotting purposes.

In this section, the capabilities and organization of the SMLEXC code are described first. Thereafter, the mechanics of data transfer between various codes, compilation and linking, system specific calls, array sizes, and units in the SMLEXC code are discussed. Finally, the SMLEXC software limitations, hardware requirements and installation procedures, hints for code maintenance, verification, benchmarking and configuration control, and user support information are provided.

3.1 SMLEXC CODE CAPABILITIES

To construct CDFs and PDFs for failures of multiple waste packages arranged in a simulated square repository using the SMLEXC code, each individual code contributes the following:

(i) The executive code reads in the key input file, smlexc.inp, which contains parameters such as the input and output filenames of other codes and the total number of sampling vectors. In addition, it controls the execution for the rest of the codes and performs simple CDF analyses for event times collected.

(ii) The LHS code performs the key probabilistic analysis for the SMLEXC code. It is capable of sampling data from both correlated or independent random variables according to the defined statistical distributions file, smllhs.dat, which is read in by the LHS code.

(iii) The SCCEXP code maps the LHS sampled data into a SCCEX code baseline input file, sccex.inp, and constructs a SCCEX code readable input file, scpscc.dat, according to a mapping file, scpmap.dat, which is read in by the SCCEXP code.

(iv) For each calculation cell, the SCCEX code performs one major deterministic analysis to produce time-to-wetting and time-to-failure results. This deterministic analysis calculates the average repository temperature, the evaporation rate of pure water from the container surface, corrosion rates, release rate, etc. Because of the complexity of computation, each SCCEX code calculation needs more time than other codes. When more vectors and cells are considered, total computational effort becomes excessive; therefore, an advanced probabilistic analysis approach, instead of the LHS approach, should be used.

(v) The POSTDD code is designed to be an interactive code (see Appendix A). This code requires the user to input the key parameters for constructing CDFs and PDFs from the calculated wetting and failure data. These key parameters are times, cells, and cumulative number of failures.

Version 1.0 of SMLEXC does not include any code for sensitivity/uncertainty analyses; these analyses are performed externally by post-processing SMLEXC output. As discussed in the report entitled "Substantially Complete Containment"—Example Analysis of a Reference Container (Cragnolino et al., 1994), several computations were made to compare different design scenarios which include different materials and environmental situations. In addition, several uncertainty analyses were also performed to understand the impact of some random properties.

3.2 SMLEXC CODE ORGANIZATION

The current SMLEXC code flow diagram is presented in Figure 3-1. As shown in Figure 3-1, the executive code directs data flow between different subprocesses and controls their execution. The executive code uses a dedicated subroutine to handle the setup and initiation of each subprocess. The subprocess is the result of a FORTRAN CALL statement invoking a utility routine, which is specific to the operating system. This CALL starts the subprocess. The execution of the executive is suspended while the subprocess is executing. Control is returned to the executive at the end of the execution of the subprocess execution.

All functions used in the executive code are shown in Table 3-1. Detailed discussion of the LHS code can be found in a report by Iman and Shortencarrier (1984). Detailed discussion for the SCCEX code can be referenced in a report by Cragnolino et al. (1994) and the SCCEX code user's manual by Torng et al. (1994). Both the SCCEXP code and the POSTDD code are not discussed in details because both codes have simple objectives and are relative simple and small.

3.3 DATA HANDLING AND CONTROL

The SMLEXC I/O files generally can be divided into three different types: (i) input; (ii) temporary; and (iii) output. For each code, one or more input files are required. Some input files are not user-controllable and are actually temporary files created by the SMLEXC system.

The temporary files are created for the purpose of transmitting control parameters to a code. These files typically contain global parameters that can override the parameters read from the input file of the code. The temporary files are generated by the SMLEXC code. In general, the user does not need to prepare or make arrangements for the temporary files other than preventing possible file name conflicts in the default directory. The output files include the output files from every code, error log files, and specially formatted files for external utilities, such as the TECPLOT graphics utility (TECPLOT, 1990). The names of most files, except the temporary files, can be read as part of the input. The format-free input process of the executive code is based on the use of keywords. Keywords used to read the names of the files are discussed in Section 4.

3.4 COMPILATION, LINKING, AND RUNNING THE SMLEXC CODE

A makefile is used for compiling and linking on a SUN computer system. In Table 3-2, an example makefile for creating the executable code, smlexc.e, for the executive code is shown.

Note that, in the makefile, -w is used to suppress the warning message, -o is used to name the executable file, smlexc.e, -ldl is used to access Fortran intrinsic functions that are contained in SUN shared libraries, -c is used to compile without linking, and -xl is used to accept the backslash (\) character

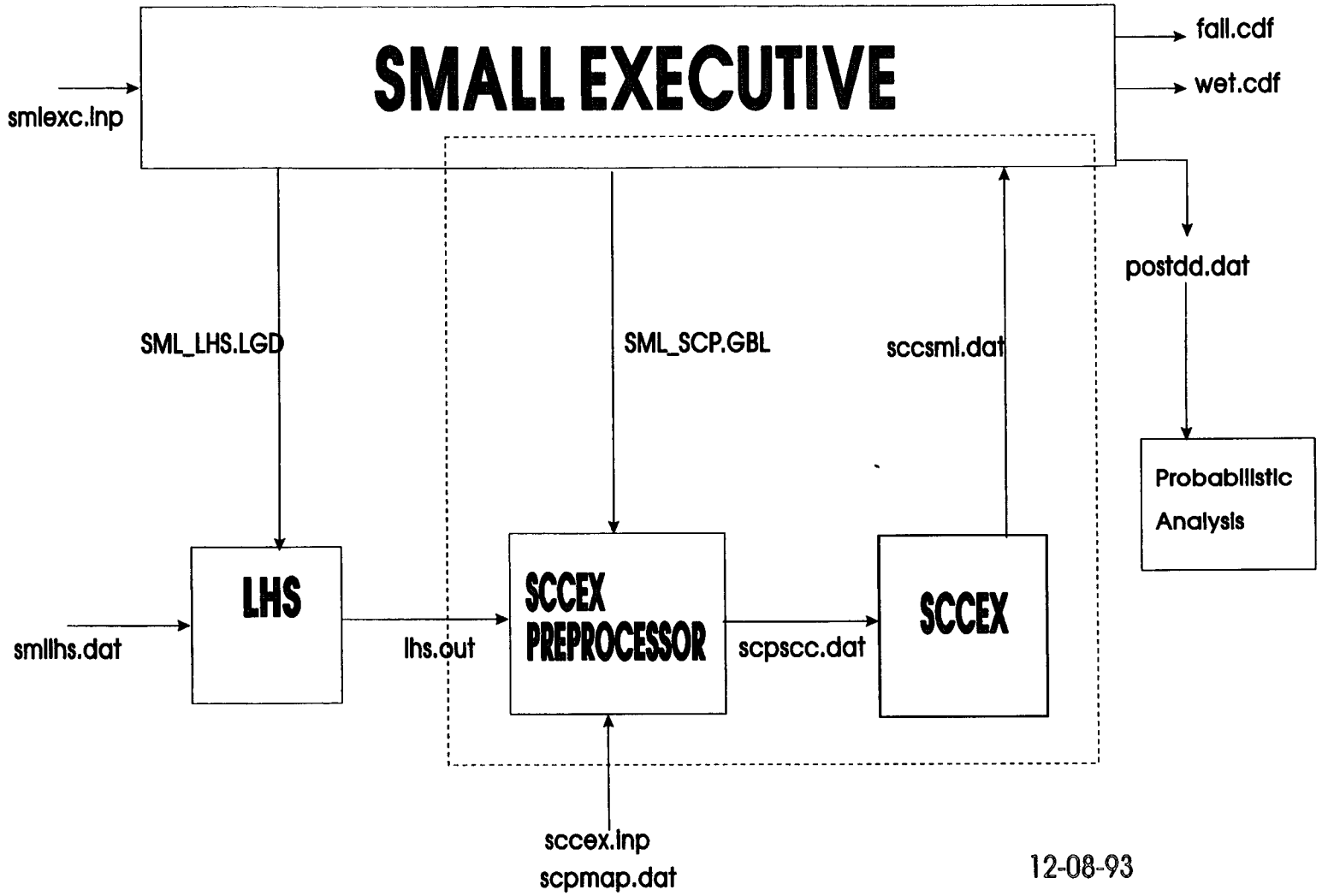


Figure 3-1. Diagram showing the data flow and the execution dependencies of the SMLEXC system and the input/output file names

Table 3-1. Listing of subroutines and their functions within the executive code

Subroutine	Function
blocks	Generates block letters for the first 18 letters of the input string and writes them to the FORTRAN logical unit given in the first argument
calcrd	Reads the calculation section keyword input for the main SMLEXC module
center	Centers an input string in an output string with blanks on both sides
cnsqnc	Calls the various executive routines for the consequence modules for the current vector
comprs	Compresses a character string by compressing all successive blanks to the maximum allowed as specified by the argument list
dmpfil	Dumps an ASCII file to the supplied FORTRAN logical unit
errmsg	Writes the string argument to the error file
esort	Sorts the "epasum" array and its corresponding probability array
init	Initializes the "cumrel" array to zero
lhsexc	Sets up and controls the environment for the LHS subprocess
logmsg	Writes the string argument with a time tag to the SMLEXC log file
opnfil	Opens a formatted file on the requested logical unit with the arguments supplied in the argument list
opninp	Opens the SMLEXC input file. This file contains free-form input suitable for reading by routines in the RDFREE library
opnlog	Opens the SMLEXC log file. The log file keeps a log of the major events in the running of the SMLEXC process
opnout	Opens the SMLEXC output file
outfmt	Writes the requested intermediate arrays to formatted disk files
output	Writes the requested intermediate arrays to the output file
paghdr	Outputs a form feed for a new page and prints a header with the run title
pagttl	Outputs a page title for a new page
parinf	Puts both Exec and RDFREE code PARAMETER items on the second title page
rdinhd	Reads the header section keyword input from the input file
scdfwr	Opens and writes the scenario CCDF file
scninp	Reads the keyword input for the main SMLEXC module
shsort	Performs a shell sort of the input array into ascending numerical order
titpag	Puts the program title page on the output device with block letters
smlexc	Serves as the main driver for the SMLEXC system

Table 3-2. Example makefile for the executive code

```
#
HOME = /home/scratchy/janetzke
smlexc.e:       smlexc.o
        f77 -w -o smlexc.e        smlexc.o -ldl
smlexc.o:       SMLEXC.F
        f77 -w -c -o smlexc.o  SMLEXC.F
SMLEXC.F:       smlexc.pre
        $(HOME)/prefor/prefor.e     < PREFOR.smlexc
```

in a FORTRAN character constant. The PREFOR.smlexc file contains the input and the output for the prefor.e code. It is also important to keep track of the path where the files are located.

To activate this makefile, the user types "make" in the same directory where the makefile is located. The same procedure can be applied for the rest of the codes to construct the executable codes. To run the SMLEXC code on a SUN computer system, the user types "smlexc.e".

3.5 SYSTEM-SPECIFIC CALLS IN THE SMLEXC CODE

The SMLEXC computer code is configured for the SUN computer system. The following are nonstandard system calls used in the SMLEXC system: (i) CALL fdate(), which is used to obtain the calender date for the SMLEXC code; (ii) dtime(mytime), which is used to record the computation time for the SCCEX code; and (iii) call ieee_flag(), which, in turn, is used to eliminate unnecessary system-related outputs.

3.6 ARRAY SIZES IN THE SMLEXC CODE

The storage size of the program is greatly affected by the size of the FALTYP, LCHTYP, and WETTYP arrays. These arrays have three dimensions: (MAXARE, MAXVEC, and MAXSCN) where,

MAXARE [maximum number of areas (cells) in repository] = 1024 (default)

MAXVEC (maximum number of vectors) = 100 (default)

MAXSCN (maximum number of scenarios) = 1 (default)

All of these may have a minimum setting of 1.

3.7 UNITS IN THE SMLEXC CODE

All units in SMLEXC are metric [i.e., meter (length), kilogram (mass), and year (time)], except for initial inventory, which is in Kilo Metric Tons Heavy Metal.

3.8 SOFTWARE LIMITATION

When the SMLEXC system is invoked, it assumes the existence of various input files and the absence of important files that are not overwritten. The analyst should move all important output files to another directory before running the SMLEXC system since the executive code will halt if it attempts to overwrite them during execution. The files assumed to be important by SMLEXC are, 'smlexc.log', 'smlexc.out', 'SCN_TLY.DAT', and 'SCN*'. A file assumed important by the LHS code is 'lhs.out'.

To avoid potential problems, a shell script called "clean" will eliminate those important files that will stop the SMLEXC code execution. The shell script, "clean," is shown in Table 3-3.

Table 3-3. Example file "clean"

```
#!/bin/csh
/bin/rm -f *_TLY*
/bin/rm -f lhs.out
/bin/rm -f smlexc.log
/bin/rm -f smlexc.out
```

3.9 HINTS FOR CODE MAINTENANCE

Code maintenance is provided by using a FORTRAN Preprocessor (preFOR, Janetzke and Sagar, 1991a), which is written in standard FORTRAN 77 language. It is a preprocessor for FORTRAN codes, and its use can provide some flexibility in developing such codes. Included in preFOR are various utility codes to find the length of strings, fix tabs, perform I/O, etc. There is only one parameter in this code that an individual user may have to change—the parameter MXCOM defines the maximum number of temporary files allowed to be generated by preFOR and may be changed to something other than 250 (default). This parameter declares the array space for the temporary file names of the code fragments. If the number of code fragments exceeds MXCOM, the parameter should be increased to match the number of fragments.

When invoked, preFOR prompts for an input file name. The input file is a source code in which preFOR commands have been embedded. The program then prompts for an output file name. The output file produced by preFOR is a standard FORTRAN file that can be compiled. Details of the preFOR commands can be found in Janetzke and Sagar (1991a).

The source code for the SMLEXC includes a number of the preFOR commands. Therefore, before compilation, the preFOR utility should be invoked to produce a standard FORTRAN file.

3.10 HARDWARE REQUIREMENTS AND INSTALLATION PROCEDURES

Typically, each subprocess code is stored in its own subdirectory, with a separate subdirectory for the SMLEXC. The primary source files are maintained as FORTRAN preFOR preprocessor input files, except the SCCEX source code files. Once the preFOR files are copied to disk, an operating system procedure can be written to automatically generate the executable files. Each code will then have an

executable file in its subdirectory. These are the codes that are spawned by the executive code during execution.

3.11 VERIFICATION, BENCHMARKING, AND CONFIGURATION CONTROL

The SMLEXC verification procedure consists of checking the input data values transferred to the subprocess via the global data files. These files are the only control the executive has on the subprocesses. The information is easily checked since it is written to an external ASCII file and can be inspected after the run is complete.

Because the subprocesses are able to run in standalone mode or as part of the SMLEXC system, this permits the output from both modes to be compared using a differencing utility to ensure common results. By using a debugger, the user can compare the voluminous data returned by the subprocesses to the executive with the results obtained from the standalone mode. In addition, the debugger is used to check the global data values after the global data file is read by the codes. The standalone version of the subprocesses are considered the benchmark version.

The SMLEXC code is managed under procedures set out in the CNWRA Technical Operating Procedure (TOP)-018. The production of this user's manual is one of the requirements of TOP-018.

3.12 USER SUPPORT

For technical assistance contact:

Ron Janetzke
Center for Nuclear Waste Regulatory Analyses
Southwest Research Institute
P. O. Drawer 28510
San Antonio, TX 78228-0510
e-mail: rjanetzke@swri.edu
Tel: (210) 522-3318

Tony Y. Torng
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78228-0510
e-mail: ttorng@SwRI.edu
Tel: (210) 522-5497

4 DESCRIPTION OF THE SMLEXC CODE INPUT FILES

The input files used in every SMLEXC code are discussed in this section. The output files will be discussed in Section 5. The temporary files, which include the files generated by the SMLEXC system for the purpose of transmitting control parameters to a code, are also discussed. These temporary files typically contain global parameters (parameters that are common to more than one code), which can override the parameters read from the input file of the code. In general, the user does not need to prepare or make arrangements for the temporary files, other than preventing possible file name conflicts in the default directory.

In Subsection 4.1, the parameters included in input files for every code in the SMLEXC system are explained or referenced to other reports.

4.1 THE EXECUTIVE CODE INPUT FILE

The input to the executive code is an ASCII data file that is created using an editor and contains a sequence of appropriate keywords. These keywords and the numeric and alphanumeric data associated with them are explained in detail in this section. Section 3.7 discusses the appropriate units for input values.

The input file for the executive code consists of three sections: (i) header information, (ii) scenario data, and (iii) output instructions. The header section contains two data lines. The first line specifies the title of the problem, and the second line specifies the identification of the analyst. The scenario section contains the specific global and code information for the run. The output instructions that specify the disposition of output data items are given in the output instruction section.

The construction of the input file for the executive code is based on the format-free RDFREE (Janetzke and Sagar, 1991b) library of subroutines. Even though the input is referred to as format-free, there are, nevertheless, a few rules to follow when composing an input line. These rules are much less restrictive than the usual FORTRAN input statements and can be found in the RDFREE Utility User's Guide (Janetzke and Sagar, 1991b). The RDFREE set of subroutines is used to scan the input for recognizable words and numbers and assigns the appropriate values to the internal variables.

4.1.1 Description of The Executive Code Input File: smlexc.inp

In Table 4-1, an smlexc.inp example input file for 304L material with intermediate initial power density case is shown. Note that the keywords are identified in bold and uppercase. These need not be entered in uppercase in the actual input file.

4.1.2 Detailed Description of smlexc.inp Keywords

This subsection defines the smlexc.inp keywords in the same sequence as shown Table 4-1. The keyword characters, which appear in lower case on the following pages, may be omitted since the input routine only recognizes the first six characters of a word. The modifiers are grouped under their respective keywords in alphabetical order. Note that all logical records must end with a special symbol, which is the vertical bar (|).

Table 4-1. smlexc.inp example

```
TITLE: SMLEXC, 100 vectors, intermediate, 304L production run II.
ANALYSt 'Ron Janetzke'
|
\\ Scenario section.\\
SCENARio                        \ Begin first scenario.
        name 'SCN_01'           \ Scenario name.
        output 'name_1.cum'     \ Scenario output file name.
        ccdf   'sccdf.dat'      \ Produce CCDF output.
        probability 1.0         \ Scenario probability.
        start time=0            \ Simulation start time.
        stop time=1000          \ Simulation stop time.
|
LHS                             \ Specify LHS parameters.
    vectors = 100               \ Number of vectors for this scenario.
    input '../lhs3/smllhs.dat' \ LHS input file name.
    output 'lhs.out'            \ LHS output file name.
|
SCCEXP
     input = 'sccex.inp'
     lhsmap = '../sccexp/scpmap.dat'
     output = 'scpscc.dat'
|
SCCEX
|
RUN
|
SCENARio end
|
\\ Calculation section. \\
OUTPUT
      ccdf
|
END|
```

The descriptions of keywords are given in five parts: (i) PURPOSE, (ii) SYNTAX, (iii) MODIFIERS, (iv) COMMENTS, and (v) EXAMPLE.

The PURPOSE part gives the section (either header, scenario, or output) in which the keyword may appear, as well as a functional description of the keyword, indicating the action implied by its use. The SYNTAX part shows the general form of the input record along with the position of any data fields. The keyword or modifier is given in uppercase, and the generic data identification is given in lowercase, if present. The MODIFIER part gives the meaning of the modifiers and any data associated with them. The data fields may be one of two types: (i) string or (ii) numeric. The numeric data values can be input

in any form (integer, fixed point, or exponent) and in any column following the keyword or modifier. String values must be enclosed in single quotes ('). The COMMENTS part gives further details of KEYWORD and data fields and hints on their usage. The EXAMPLE part gives an example of a logical record that uses the keyword.

According to Section 4.1.1 example input, keywords in the following sequence are discussed in Tables 4-2 to 4-9: TITLE, ANALYSt, SCENARio, LHS, SCCEXP, SCCEX, RUN, and OUTPUT.

Table 4-2. TITLE

PURPOSE	This header section keyword provides the title for the run and appears on the top of each output page. The title string is limited to 74 characters. This parameter is required to satisfy quality assurance concerns.
SYNTAX	TITLE s1 s1: An unquoted string.
MODIFIERS	None
COMMENTS	Note that the vertical bar is not required to terminate the title. This keyword always takes the next 74 characters on the line as the title string.
EXAMPLE	TITLE: Test #1.

Table 4-3. ANALYSt

PURPOSE	This header section keyword identifies the analyst associated with the current input file. As a required parameter, it enhances the quality assurance feature of the SMLEXC system. This data item appears in the SMLEXC printed output.
SYNTAX	ANALYSt s1 | s1: Analyst name.
MODIFIERS	None
COMMENTS	Only one string field is allowed, so include the first name, last name, and any initials in the string.
EXAMPLE	ANALYSt 'Ron Janetzke' |

Table 4-4. SCENARio

PURPOSE	This scenario section keyword indicates the end of parameter input for a given scenario and requests the modules to be executed in the order in which they were defined in the input file. This keyword introduces a logical record of modifiers that defines the parameters for a given scenario. Some of the modifiers define data values that are passed to some or all of the modules. Scenario name and probability are specified as well as CCDF plot files.
SYNTAX	SCENARio CCDF s1 LHS s2 NAME s3 OUTPUT s4 PROBABILITY n4 START n5 STOP n6 | OR: SCENARio END |
MODIFIERS	CCDF s1: Name of the file to be used for the scenario CCDF plot data. The CCDF generated is for the scenario only and not for the entire run. This modifier must be present for each scenario for which a plot file is desired. END: This modifier is used alone with no others. It terminates the scenario section of the SMLEXC input file and must appear after all other scenario input commands. The calculation section is entered immediately upon the receipt of the END modifier for the SCENARio keyword. LHS s2: Name of the LHS output file that holds the sampled data for this scenario. NAME s3: A character designation for this scenario. OUTPUT s5: This modifier identifies the name of the file to be used for storing the CUMREL array for the scenario. PROBABILITY n18: Scenario probability. If a scenario CCDF is requested, this should be 1. START n19: Simulation start time. STOP n20: Simulation stop time.
COMMENTS	The END modifier is used on a mutually exclusive basis with all of the other modifiers. In Version 2.0 of SMLEXC, canister length is assumed to be the same in all cells of the repository.
EXAMPLE	name 'SCN_01' \ Scenario name. output 'name_1.cum' \ Scenario output file name. ccdf 'sccdf.dat' \ Produce CCDF output. probability 1.0 \ Scenario probability. start time=0 \ Simulation start time. stop time=1000 \ Simulation stop time. |

Table 4-5. LHS

PURPOSE	This scenario section keyword is used to specify the parameters that are to be passed to the LHS module. The data that is passed from the executive code to the LHS generally consist of the names of the I/O files and the number of sample vectors to be generated.
SYNTAX	LHS INPUT s1 OUTPUT s2 VECTORS n1 |
MODIFIERS	INPUT s1: The name of the LHS input file. OUTPUT s2: The name of the LHS output file with the sampled data. VECTORS n1: The number of vector data sets to generate. This value overrides the value for the number of observations (NOBS) parameter in the LHS input file.
COMMENTS	Since the LHS parameters can affect all modules, they should appear before them in the input file. Only one number should follow the keyword; any numbers appearing after the first number will be ignored. This number must not exceed MAXVEC, because this is the FORTRAN parameter that controls the size of many of the internal data arrays. MAXVEC can be set to any value compatible with the host machine.
EXAMPLE	LHS vectors 50 input = 'lhs.inp' output = 'lhs.out' |

Table 4-6. SCCEXP

PURPOSE	This scenario section keyword schedules the SCCEXP preprocessor for execution in the executive sampling loop.
SYNTAX	SCCEXP INPUT s1 LHSMAP s2 OUTPUT s3 |
MODIFIERS	INPUT s1: The name of the default parameter file for SCCEX. This file could also be used as input to SCCEX in stand-alone mode. LHSMAP s2: The name of the LHS map file that contains the indices for the sampled parameters. OUTPUT s3: The name of the SCCEXP output file that will become the new SCCEX input file.
COMMENTS	All file names are required and may include the path specification.
EXAMPLE	SCCEXP INPUT 'sccex.inp' LHSMAP 'scpmap.dat' OUTPUT 'scpscc.dat' |

Table 4-7. SCCEX

PURPOSE	This scenario section keyword schedules the SCCEX module for execution in the main sampling loop of the executive.
SYNTAX	SCCEX |
MODIFIERS	None
COMMENTS	Control and input parameter specification for this module is handled by the executive and the SCCEXP preprocessor, so no modifiers are required from the user.
EXAMPLE	SCCEX |

Table 4-8. RUN

PURPOSE	This scenario section keyword starts the execution loop of all modules scheduled for this scenario.
SYNTAX	RUN |
MODIFIERS	None.
COMMENTS	The use of the RUN keyword signals the end of the specification of a given scenario. For single scenario runs, the SCENARIO END command usually follows this logical record immediately.
EXAMPLE	RUN |

Table 4-9. OUTPUT

PURPOSE	This output section keyword specifies that formatted output files of internal arrays are to be generated by the SMLEXC module. A CCDF is generated and written to a formatted plot file containing the total CCDF for the run.
SYNTAX	OUTPUT CCDF |
MODIFIERS	CCDF This modifier requests the generation of an ASCII file containing the total CCDF for the run.
COMMENTS	None.
EXAMPLE	OUTPUT ccdf |

4.2 LHS CODE INPUT FILES

Two input files are required by the LHS code in order to sample data and create an output file, lhs.out. The first input file, smllhs.dat, contains the statistical information for the random variables used in the LHS code. The second input file required is a temporary file created by the executive code called SML_LHS.LGD, which transfers global data from the executive code to the LHS code.

4.2.1 Description of the LHS Input File: smllhs.dat

The method for constructing the LHS input file is described in the LHS documentation (Iman and Shortencarrier, 1984). Please refer to this document for definitions of FORTRAN variable names used below. Compared to the format-free style of the SMLEXC input file described previously, the LHS input file is of fixed-format type.

As shown in Table 4-10, an smllhs.dat example file, which includes a total of 49 parameters, is shown. The first line in the LHS input provides the title of the problem. This title will be written in the LHS output file. Preferably, this title should be the same as the one in the SMLEXC input file. The remaining lines in the LHS input file consist of keywords (beginning in the first column) followed on another line by numeric values in fixed format.

Note that while using the LHS as part of the SMLEXC system, the number of observations (NOBS) keyword may be omitted if the VECTORS modifier is used in the SMLEXC input file (see previous subsection). If both NOBS and VECTORS are used, the VECTORS value obtained from the executive code will overwrite the NOBS value. This is accomplished through a slightly modified input section in the LHS program that looks for this global parameter in the LHS global data (SML_LHS.LGD) file. The value used for the number of repeated samples (NREPS) should be always set to 1. For details of statistical distributions that can be invoked, the documentation of LHS should be consulted. Note that variable names are not required, but are included to reflect the names used in the codes. Notice also that some variables are followed by additional text for clarity. Since the correlation matrix has not been defined in Table 4-10, these random variables are uncorrelated.

Note that several explanations after the exclamation point are added for clarification purpose. However, in the original file, these explanations after the exclamation point will not be seen. The same format can be seen in the following subsections.

4.2.2 Description of the LHS Input File: SML_LHS.LGD

This SML_LHS.LGD file is a temporary file created by the SMLEXC program and is not supplied by the analyst. It contains the values of variables that have been selected to overwrite the LHS input values. These include the I/O file names and the number of vectors requested, as shown in Table 4-11.

4.3 SCCEXP CODE INPUT FILES

Four input files are required by the SCCEXP code in order to map the LHS sampled output file, lhs.out, into an input file for the SCCEX code. To perform the mapping job, the executive code creates a temporary file called SML_SCP.GBL to control this mapping process. In this file, it indicates the

Table 4-10. smllhs.dat example

```
TITLE - LHS input file for SCCEX parameters. Ecorr added. 5-24-93
RANDOM SEED   -169
NOBS  2
NREPS  1
UNIFORM          age     [years]  ; Age of fuel.
10.0   60.0
UNIFORM          tkr     [J/yr/m/K]; Thermal conductivity for rock.
4.7e7  6.8e7
UNIFORM          tkp     [J/yr/m/K]; Thermal conductivity for packing.
2.7e7  4.8e7
UNIFORM          rhocpr  [J/m**3/K]; Density*heat_capacity for rock.
2.e6  3.e6
LOGUNIFORM         taur    [-]       ; Tortuosity for rock.
0.01   1.
UNIFORM          phir    [-]       ; Porosity for rock.
0.05   0.2
UNIFORM          delta   [m]       ; Water film thickness.
1.0e-3  3.0e-3
UNIFORM          salt    [kg/m**3] ; Salt concentration.
0.05   0.2
UNIFORM          scale   [kg/m**3] ; Scale concentration.
0.0    0.5
UNIFORM          bfact   [-]       ; Vapor pressure lowering scale factor.
0.7    1.3
UNIFORM          dsalt   [kg]      ; Initial salt.
0.0    5.0
UNIFORM          dscale  [kg]      ; Initial scale.
0.0   15.0
LOGUNIFORM         dfrac   [m**2]    ; Funnel area
0.1    10.0
UNIFORM          slice   [m]       ; Umbrella edge thickness.
1.0    30.0
UNIFORM          tenhan  [C]       ; Umbrella edge temperature.
90.0   98.0
UNIFORM          tslop   [C]       ; +/- range of 'tenhan'.
0.1    5.0
UNIFORM          betaox  [-]       ; oxygen beta kinetics parameter.
0.25   1.5
UNIFORM          betahy  [-]       ; water beta kinetics parameter.
0.25   0.5
UNIFORM          rkox    [C*m/(mole*yr)]; constant for oxygen reduction.
3.e6   3.e14
UNIFORM          rkhy    [C/(m*m*yr)]; rate constant for water reduction.
3.2e-4   3.2e4
CONSTANT          Gox      [J/mol]   ; Activation energy for oxygen rate.
100000.
```

Table 4-10. smllhs.dat example (cont'd)

```
CONSTANT        Ghy      [J/mol]   ; activation energy for water rate.
20000.
UNIFORM         curair   [C/(yr*m*m)]; current density for air/steam.
6.3    6.3e3
UNIFORM         curact   [C/(yr*m*m)]; active current density.
6.3e5  6.3e8
UNIFORM         aa(1)    [C/(yr*m*m)]  ; passive current density coefficient.
6.3e3  6.3e5
CONSTANT        aa(2)    [-]       ; passive current density coefficient.
0.0
CONSTANT        aa(3)    [-]       ; passive current density coefficient.
0.0
UNIFORM         CEcrit   [-]       ; coefficient for critical corrosion potential
0.     1.
CONSTANT        bb(2)    [-]       ; coefficient for critical corrosion potential
0.0
CONSTANT        bb(3)    [-]       ; coefficient for critical corrosion potential
0.0
CONSTANT        bb(4)    [-]       ; coefficient for critical corrosion potential
0.0
CONSTANT        bb(5)    [-]       ; coefficient for critical corrosion potential
0.0
CONSTANT        bb(6)    [-]       ; coefficient for critical corrosion potential
0.0
CONSTANT        clrat    [-]       ; nitrate/cloride ratio.
1.0
UNIFORM         taus     [-]       ; tortuosity correction for scale.
0.1    1.0
UNIFORM         spor     [-]       ; scale porosity.
0.2    1.0
CONSTANT        ndrip    [count]   ; Number of drip rate values.
6.
CONSTANT        qtime(1) [years]   ; Time for qdrip.
0.
CONSTANT        qtime(2) [years]   ; Time for qdrip.
100.
CONSTANT        qtime(3) [years]   ; Time for qdrip.
400.
CONSTANT        qtime(4) [years]   ; Time for qdrip.
3000.
CONSTANT        qtime(5) [years]   ; Time for qdrip.
8000.
CONSTANT        qtime(6) [years]   ; Time for qdrip.
100000.
```

Table 4-10. smllhs.dat example (cont'd)

```
LOGNORMAL          qdrip(1) [m/yr]    ; Infiltration.
1.0E-4  5.0E-3
LOGNORMAL          qdrip(2) [m/yr]    ; Infiltration.
1.0E-4  5.0E-3
LOGNORMAL          qdrip(3) [m/yr]    ; Infiltration.
1.0E-4  5.0E-3
LOGNORMAL          qdrip(4) [m/yr]    ; Infiltration.
1.0E-4  5.0E-3
LOGNORMAL          qdrip(5) [m/yr]    ; Infiltration.
1.0E-4  5.0E-3
LOGNORMAL          qdrip(6) [m/yr]    ; Infiltration.
1.0E-4  5.0E-3
```

Table 4-11. SML_LHS.LGD example

```
   SMLEXC temporary file for LHS global parameters.
   ../lhs3/smllhs.dat  ! file name and location
   lhs.out  ! output file name
100 ! one hundred set of data to be sampled
```

baseline SCCEX code input file name, the LHS output file name with sampled data, the mapping of the LHS data to a SCCEX baseline input, and the final mapped file name. In the following subsections, the temporary file generated by the executive code, the SCCEX baseline input file, the LHS output file, and the mapping file are discussed. The final mapped file is an output file which will be discussed in the Section 5.3.

4.3.1 Description of the SCCEXP Input File: SML_SCP.GBL

In Table 4-12, an SML_SCP.GBL example file is shown.

Table 4-12. SML_SCP.GBL example

```
TITLE: SMLEXC temporary file for SCCEXP global parameters.
VERSION: 05-18-93; Generated by SMLEXC.
'sccex.inp                ' ! SCCEX baseline input
'lhs.out                  ' ! LHS output file name with sampled data
'../sccexp/scpmap.dat       ' ! mapping LHS data to SCCEX baseline input
'scpscc.dat               ' ! new mapped file name
   100 ! perform one hundred times
```

4.3.2 Description of the SCCEXP Input File: sccex.inp

In Table 4-13, an SCCEX code baseline input data file, sccex.inp, is shown. This file controls many deterministic parameters which will not be overwritten by the sampled data. Within these parameters, the end simulation time and the total number of cells within the repository are very important factors that should be carefully considered.

4.3.3 Description of the SCCEXP Input File: scpmap.dat

There are some variables in the sccex.inp file that are also specified in the LHS input file. A map file (scpmap.dat) is provided by the analyst, giving the correspondence between such variables in the LHS output file and the variables in the input file of the SCCEXP subprocess. This file additionally serves to indicate to the subprocess the variables to be overwritten. The map file indicates the location of variables that are sampled by the LHS code and whose sample vectors therefore exist in the LHS output file. An scpmap.dat example file is shown in Table 4-14.

4.3.4 Description of the SCCEXP Input File: lhs.out

In Table 4-15, an lhs.out example file is shown. This file contains 100 sets (or realizations) of data, and each set contains a total of 49 data which will be mapped into an SCCEX input file.

4.4 SCCEX CODE INPUT FILE

The SCCEX code is invoked by the SMLEXC system and uses the input file (scpscc.dat) prepared by the SCCEXP code. The program completes its execution by producing the outputs as described in the SCCEX user's guide (Torng et al., 1994).

4.4.1 Description of the SCCEX Input File: scpscc.dat

This file is essentially the same file as sccex.inp, which was discussed in Section 4.3.2. However, some parameters have been replaced by the sampled data from lhs.out (e.g., the age of fuel). In Table 4-16, an scpscc.dat example file is listed.

4.5 POSTDD CODE INPUT FILE

The POSTDD code is developed to postprocess the system analysis data file, postdd.dat. As discussed in Section 2, this code is not directly controlled by the executive code. This code is currently run as an independent code. A detailed user instruction for using the POSTDD code is summarized in Appendix A.

4.5.1 Description of the POSTDD Input File: postdd.dat

The input file, postdd.dat, is used to collect the most valuable information for each SMLEXC system code analysis. This information includes the weight factor, time-to-failure, time-to-wetting, and time-to-leachate for every calculation cells in every simulation vectors. These valuable data are then analyzed by using a postprocessing code called POSTDD to produce data for probabilistic assessment.

Table 4-13. sccex.inp example

```
This is the Brine or SCCEX code input file
1000.        ! end simulation time
0., 0., 0.,  256 !container x,y,z location, number of cells
4.76, 0.33, 0.34, 5.0     ! rlen, r1, r2, r3
0.01     ! container thickness in meters
Temperature Parameters
4.44473e8   ! initial power density in J/yr/m^2
70000., 2.3, 10.!metric ton initial heavy metal total, MTIHM can, age of fuel
5.68e7, 3.787e7, 3.26e6! J/yr/m/K thermal conductivity for rock, packing, can
2.4e6, 1.2e6, 1.2e6  ! rho*Cp in J/m^3/K, rock, packing, can
23., 4.0  ! initial temperature (Celsius) and vertical extent of repository
Evaporation Parameters
962.,  59., 1.0, 1.0  ! D (m^2/yr), tdiff (C), tau rock, tau packing
0.12, 0.12, 2.e-3  ! phi rock, phi pac, water film thick (m)
6       ! ndrip, the number of time,drip rate pairs
0.,      1.e-3 ! time in yr, drip rate in m/yr
100.,    1.e-3
400.,    1.e-3
3000.,   1.e-3
8000.,   1.e-3
100000., 1.e-3
0.0613, 0.2121 !  salt (Kg/m^3), scale (Kg/m^3)
1.0  ! bfact, scale factor for vapor pressure lowering relative to NaCl
0.1, 0.0  ! initial salt (Kg), initial scale (Kg)
1, 1.0   ! idrip (0=> no umbrella, 1=> umbrella effect), qfrac in m^2
20., 95, 2.  ! slice, tenhan, tslop
0.895   ! atmospheric pressure (atm)
1.e-3, 1.39e-4  ! alt, lambda
1.e-3, 1.e0, 1 ! dt ini, dtmax, isolve 0->full, 1-> local equilibrium
1.e-3, 1.e-3  ! relative error, absolute error
10., 0.01 !Evap/Qin factor, Vwater/Vcrit to switch from local equilibrium
Corrosion Parameters
0.75, 0.5 ! betaox, betahy
3.e10, 3.2, 100000., 20000. ! rkox, rkhy, Gox, Ghy
6.3e2, 6.3e6 ! curair, curact
6.3e4, 0.0, 0.0 ! aa(1), const., const.
0.0, 0.0, 0.0, 0.0, 0.0, 0.0  ! CEcrit,Counter,const.,const.,const.,const.
1.0, 7.0  ! nitrate/cl ratio, reference pH
0.3, 0.5 ! tortuosity in scale, porosity in scale
```

Table 4-14. scpmap.dat example

```
TITLE: SCCEXP map file for LHS variables.
VERSION: 05-17-93; Original with 6 element qtime & qdrip arrays.
32 variables = 28 scalars + 4 arrays = 49 data values.
VARIABLE INDEX COUNT
'age'    1    1 ! age variable position 1 has one data
'tkr'    2    1
'tkp'    3    1
'rhocpr' 4    1
'taur'   5    1
'phir'   6    1
'delta'  7    1
'salt'   8    1
'scale'  9    1
'bfact'  10   1
'dsalt'  11   1
'dscale' 12   1
'dfrac'  13   1
'slice'  14   1
'tenhan' 15   1
'tslop'  16   1
'betaox' 17   1
'betahy' 18   1
'rkox'   19   1
'rkhy'   20   1
'Gox'    21   1
'Ghy'    22   1
'curair' 23   1
'curact' 24   1
'aa'     25   3 ! aa is a array with size of three
'bb'     28   6
'clrat'  34   1
'taus'   35   1
'spor'   36   1
'ndrip'  37   1
'qtime'  38   6
'qdrip'  44   6
```

Table 4-15. lhs.out example

```
     1        49 ! first data set, 49 sampled parameters
 0.5955357E+02  0.6071842E+08  0.3201504E+08  0.2056904E+07  0.5035967E-01
 0.1433044E+00  0.2962184E-02  0.1565893E+00  0.2667075E+00  0.1223796E+01
 0.3083017E+01  0.2881417E+01  0.1161487E+01  0.1933476E+02  0.9294939E+02
 0.8337170E+00  0.1435763E+01  0.3119096E+00  0.2848598E+15  0.1741379E+05
 0.1000000E+06  0.2000000E+05  0.6308075E+03  0.2276032E+09  0.1007032E+06
 0.0000000E+00  0.0000000E+00  0.2558273E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.1000000E+01  0.2868533E+00
 0.4607189E+00  0.6000000E+01  0.0000000E+00  0.1000000E+03  0.4000000E+03
 0.3000000E+04  0.8000000E+04  0.1000000E+06  0.7228362E-03  0.1275374E-02
 0.5719220E-03  0.8524279E-03  0.5524678E-03  0.9388926E-03

 ! Data sets 2-99

    100       49! data set 100, 49 sampled parameters
 0.2357107E+02  0.6470730E+08  0.4192481E+08  0.2631628E+07  0.3653558E+00
 0.1446920E+00  0.1972897E-02  0.1158106E+00  0.1426495E+00  0.1000415E+01
 0.2093332E+01  0.9154908E+01  0.6486060E+01  0.7851719E+01  0.9389057E+02
 0.3151716E+01  0.7767338E+00  0.4855712E+00  0.3591595E+13  0.2973602E+05
 0.1000000E+06  0.2000000E+05  0.2170690E+04  0.5314153E+09  0.4421945E+06
 0.0000000E+00  0.0000000E+00  0.9475294E+00  0.0000000E+00  0.0000000E+00
 0.0000000E+00  0.0000000E+00  0.0000000E+00  0.1000000E+01  0.4415553E+00
 0.2337796E+00  0.6000000E+01  0.0000000E+00  0.1000000E+03  0.4000000E+03
 0.3000000E+04  0.8000000E+04  0.1000000E+06  0.7702642E-03  0.8899244E-03
 0.5556498E-03  0.2841197E-03  0.7359567E-03  0.1351628E-02
```

Table 4-16. scpscc.dat example

```
TITLE: Standard input file for program SCCEX
 1000.0  !Simulation end time.
 0.0   0.0   0.0    256 !Container location x,y,z, number of cells.
4.76  0.33  0.34  5.00   !Cont len,rad,pack rad,outer boundary.
    1.00000E-02 !Container thickness in meters.
Temperature parameters.
    4.44473E+08 !Initial power density in J/yr/m**2.
 70000.0  2.30 23.571  !Initial MTHM,MTHM/can,age of fuel.
6.471E+07 4.192E+07 3.260E+06  !Therm cond rock, pack, can.
2.632E+06 1.200E+06 1.200E+06  !Dens*heat_cap rock, pack, can.
    23.0000    4.00000 !Inital temp(C), vertical extent of repository.
Evaporation Parameters.
 962.0   59.0   0.365  1.00  !Diff,tdiff,geom factor rock, pack.
1.447E-01 1.200E-01 1.973E-03  !Por rock, pack, water film thick.
  6 !Number of drip rate entries.
    0.0  7.703E-04  !Time(yr), drip rate(m/yr).
  100.0  8.899E-04  !Time(yr), drip rate(m/yr).
  400.0  5.556E-04  !Time(yr), drip rate(m/yr).
 3000.0  2.841E-04  !Time(yr), drip rate(m/yr).
 8000.0  7.360E-04  !Time(yr), drip rate(m/yr).
100000.0  1.352E-03  !Time(yr), drip rate(m/yr).
   0.115811   0.142650 !Concentration salt(kg/m**3), scale (kg/m**3).
 1.00041 !Scale factor for vapor pressure lowering relative to NaCl sol.
 2.09333    9.15491 !Initial salt(kg), scale(kg).
1    6.48606 !idrip=0=> no umbrella, qfrac(m**2).
7.852   93.891    3.152  !slice, tenhan, tslop(+/-tenhan).
0.895000 !Atmospheric pressure (atm).
 1.00000E-03    1.39000E-04 !Alteration rate(alt), decay rate (lambda).
1.0E-03  1.0E+00 1 !dt init, dt max, 0=>full, 1=>local equilibrium
1.0E-03  1.0E-03  !relative error, absolute
1.00E+01 1.00E-02  !Evap/Qin, Vwater/Vcrit to switch from local eq.
Corrosion Parameters
 7.767E-01  4.856E-01  !betaox, betahy
 3.592E+12  2.974E+04  1.000E+05  2.000E+04  !rkox, rkhy, Gox, Ghy
 2.171E+03  5.314E+08  !curair, curact
 4.422E+05  0.000E+00  0.000E+00  !aa(1-3)
 9.475E-01  0.000E+00  0.000E+00  0.000E+00  0.000E+00  0.000E+00  !bb(1-6)
 1.0E+00  7.0E+00  !nitrate/cl, reference pH
 4.416E-01  2.338E-01  !scale tortuosity, porosity
```

In Table 4-17, a postdd.dat example file which has only two simulation vectors and three calculation cells is listed. For a repository with three calculation cells, based on a symmetry assumption, this repository has a total of 16 cells (see Figure 2-1).

Table 4-17. postdd.dat example

```
***** SCCEX FAILURE, WETTING, LEACHATE TIME *****  ! vector 1
results
   3 ! three calculation groups
   1 ! counter
   4 ! weight factor (similar cells)
   1000, 0 ! failure time, type (0: No failure, 1: Buckling, 2: Fracture, 3: Yield, 4:
Corrosion)
   1000, 0 ! wetting time, type (0: no wet, 1: wet)
   1000, 0 ! leachate time, type (0: no leachate, 1: leachate)
   2
   8
   975,  4
   925,  1
   1000, 0
   3
   4
   950,  4
   900,  1
   1000, 0
   ***** SCCEX FAILURE, WETTING, LEACHATE TIME ***** ! vector 2 results

   3
   1
   4
   990, 0
   935, 0
   1000, 0
   2
   8
   955,  4
   905,  1
   1000, 0
   3
   4
   915,  4
   875,  1
   1000, 0
```

4.6 MAJOR INPUT FILES SUMMARY

In summary, as discussed in the previous subsections, the most important input files that the user needs to prepare or control are the following files: smlexc.inp, smllhs.dat, scpmap.dat, and sccex.inp. In addition to these files, other files are usually generated temporarily to transfer control (or global) parameters to different codes or to store the important results. The major functions for these four files are summarized in Table 4-18.

Table 4-18. Key input files summary

File Name	Major Functions
smlexc.inp (Subsection 4.1.1)	- Control input and output filenames of other modules - Control total sampling size
smllhs.dat (Subsection 4.2.1)	- Control the number of random variables - Control random variables statistics
scpmap.dat (Subsection 4.3.2)	- Control the number of random variables - Control the position of mapping
sccex.inp (Subsection 4.3.3)	- Control deterministic design parameters that will not be affected by the LHS data, especially total number of cells and total simulation time

5 DESCRIPTION OF THE SMLEXC CODE OUTPUT FILES

In this section, output files from each code in the SMLEXC code are discussed. In addition, log files, error message files, tally files for code executions, and specially formatted files for external utilities, such as the TECPLOT graphics utility (TECPLOT, 1990), are also discussed.

5.1 THE EXECUTIVE CODE OUTPUT FILES

The files that the executive code produces include two temporary files, SML_LHS.LGD and SML_SCP.GBL; one output file, smlexc.out; one log file, smlexc.log; one code execution tally file, SCN__TLY.DAT; two special formatted files for plotting purposes, fail.cdf and wet.cdf; and two results collection files, postdd.dat, which contain the most important information from the analysis, and name_1.cum, which is used to collect cumulative release values for a specific scenario. Two temporary files (SML_LHS.LGD and SML_SCP.GBL) and the summary file (postdd.dat), which are used as input files to other codes, are discussed in Subsections 4.2.2, 4.3.1, and 4.5.1, respectively. Therefore, in the following subsections, only the output file, the logfile, the error message file, the special formatted files, and the code summary file, are discussed.

In addition, two other files are created: ccdf.fmt and epasum.fmt. File ccdf.fmt contains the release CCDF information for all scenarios invoked during a given run of the SMLEXC system, and epasum.fmt contains the Environmental Protection Agency (EPA) sums. Both files are not discussed further because they are designed for debugging purposes.

5.1.1 Description of the Executive Code Output File: smlexc.out

The smlexc.out file is an output summary file for the executive code. A title is displayed in block letters followed by the software version number. Various configuration parameters are listed followed by an input echo for this run. The scenario input section of the input file is then repeated before each scenario is run. If LHS is invoked, its input file is also listed in this file.

This formatted file contains a header with quality control information; the file reproduces the input contained in the scenario and output instruction sections of the SMLEXC input file. Additionally, this file contains all FORTRAN unit 6 output, that has not been redirected to a file, from all subprocess codes for all realizations.

Other information may also be displayed depending on the settings of the xUNIT variables in the SMLEXC source code. For example, if xUNIT=6, then the error messages generated by SMLEXC are listed in the SMLEXC output file. Due to the enormous size of the file, this file is not listed in the report.

5.1.2 Description of The Executive Code Log File: smlexc.log

This file contains the log entries from the SMLEXC code "logmsg." The messages indicate the times at which certain events in the executive process took place. By examining the start times of the subprocess an estimate of wall clock time can be derived for each code.

In addition, some process failure conditions may also be indicated, such as the abnormal return of a subprocess. This formatted file contains a log of the major events during the execution of SMLEXC. The first line of this file contains the date and title. The remaining lines contain the time and a message describing the event. Events such as the opening of a file and the starting of a subprocess are included. A smlexc.log example file is shown in Table 5-1.

Table 5-1. smlexc.log example

```
17-JUN-91 SMLEXC version 2.0 process log, CNWRA:05/27/91
17:25:33 SMLEXC output file opened with name on next line.
17:25:33 USER4:[RJANETZKE.SMLEXC2]SMLEXC.OUT;27 ]
17:25:33 SMLEXC input file opened with name on next line.
17:25:33 USER4:[RJANETZKE.SMLEXC2]SMLEXC.INP;12
17:25:34 Analyst is Ron Janetzke
17:25:34 Scenario name: Name_1
17:25:34 LHS input file is [rjanetzke]lhs.inp
17:25:34 LHS output file is lhs.ou1
17:27:46 Scenario CCDF file name is SMLEXCS01.CDF
17:27:46 File on next line is opened for scenario CDF file.
17:27:46 USER4:[RJANETZKE.SMLEXC2]SMLEXCS01.CDF;1
17:28:21 Normal SMLEXC system exit.
```

5.1.3 Description of The Executive Code Error Message File: smlexc.err

This formatted file is normally empty if no errors are produced during the execution of the program. If the SMLEXC system fails for some reason, there may be one or two messages in this file that would reflect the possible trouble area. In Appendix B, a list of error messages for the SMLEXC code is shown.

5.1.4 Description of The Executive Code Tally File: SCN_TLY.DAT

This file is an execution tally file that logs the success or failure of each subprocess code execution for each vector of a scenario. The first four letters of the file name are reserved for the scenario name.

The file contains a simple 2D matrix of code versus vector number with an integer in each position indicating the status of the run. The symbol (-1) indicates that the code was not selected to run for that particular vector. The symbol (0) indicates that the run was unsuccessful. The symbol (1) indicates that the run was successful. The scheme implemented is neither comprehensive nor robust but does detect any code abnormal termination from the operating system point of view (e.g., divide by zero). The symbol SCN_TLY.DAT example file is shown in Table 5-2.

Note that explanations after the exclamation point are added for clarification purposes. However, they are not presented in the original file. The same format is used in the following subsections.

Table 5-2. SCN_TLY.DAT example

```
TITLE: Completion tally of modules vs. vector for one scenario.
  2 ; number of modules
Module names
' SCCEXP '
' SCCEX  '
  100 ; number of vectors
  1  1  -1  -1 ! module one (1: yes, -1: no), module two, three and four
  1  1  -1  -1

  ! data from 3-97

  1  1  -1  -1
  1  1  -1  -1
```

5.1.5 Description of The Executive Code CDF Plot Files: fail.cdf and wet.cdf

The names of the fail.cdf and wet.cdf files are user-supplied with the SCENARIO CCDF keyword in the SMLEXC input file. The files contain the scenario CCDF in TECPLOT format. Note that it does not contain CCDF information from more than one scenario.

These files contain sorted event times and associated probabilities for the run. These files are specially formatted for use with the TECPLOT graphics utility (TECPLOT, 1990). See the TECPLOT user's guide for more explanations. A fail.cdf example file is shown in Table 5-3.

Table 5-3. fail.cdf example

```
TITLE="Scenario CCDF for SMLEXC test run."
VARIABLES=Entry,Year,Proba,Compl
ZONE T="CCDF",F=POINT
  1    7.0556E+00  2.5000E-01  1.0000E+00
  2    6.2449E+01  2.5000E-01  7.5000E-01
  3    9.1399E+01  2.5000E-01  5.0000E-01
  4    1.0301E+02  2.5000E-01  2.5000E-01
```

5.1.6 Description of The Executive Code Scenario Output File: name_1.cum

The name of the name_1.cum file is user supplied with the SCENARIO output keyword in the SMLEXC input file. It is a scenario output file name. In Table 5-4, a name_1.cum example file is shown.

Table 5-4. name_1.cum example

```
SMLEXC Version 1.0 on Jul 14 14 at 09 1994:
SMLEXC test run.
SMLEXC scenario cumulative release data file.
Areas  Modules  Vectors
1024      2       2
      1 vector
      2 vector
```

5.2 LHS CODE OUTPUT FILES

Two files that the LHS code produces are "lhs.out" (which is also used as an input file for the SCCEXP code) and "lhsconst.dat" (which provides identification of constant data in the LHS output file). The lhs.out file is discussed in Section 4.3.4 because this file is also an input file for the SCCEXP code.

5.2.1 Description of the LHS Output File: lhsconst.dat

The lhsconst.dat file is a summary file created by the LHS program and is not supplied by the analyst. It identifies the constant data in the LHS output file (according to smllhs.dat file). As shown in Table 5-5, a lhsconst.dat example file is shown. Compare Table 4-10 and 5-5 to locate the positions of constant values.

5.3 SCCEXP CODE OUTPUT FILES

The only file that the SCCEXP code produces is called "scpscc.dat", which is also used as an input file for the SCCEX code. In Section 4.4.1, an SCCEXP code example output file is shown.

5.4 SCCEX CODE OUTPUT FILES

The files that the SCCEX code produces include simple.out, main.out, mechanical.out, release.out, summary.out, and sccsml.dat. Among these files, only sccsml.dat is read by the executive code in order to construct postdd.dat. The rest of the files, which are overwritten by the following calculations, are not discussed further. However, detailed discussions of these files can be found in the SCCEX user's manual (Torng et al., 1994). Therefore, in this subsection, only the sccsml.dat is discussed.

5.4.1 Description of the SCCEX Output File: sccsml.dat

The format of this file is the same as in postdd.dat file (see Subsection 4.5.1). However, this file contains data for only one vector calculation and postdd.dat contains more than one vector calculation (depends on how many vectors are required). In Table 5-6, an sccsml.dat example file with a total of 256 cells (which is equivalent to a total of 36 calculation cells) is used for demonstration.

Table 5-5. lhsconst.dat example

```
TITLE:  LHS constant locater file.
VERSION:  File format implemented on Jan. 25, 1993.  rwj
    17  ; number of constants.
21
22
26
27
29
30
31
32
33
34
37
38
39
40
41
42
43
```

5.5 POSTDD CODE OUTPUT FILES

To extract the most important data files using the POSTDD code, the input file, postdd.dat, must be used. Given an input file name called "filename.dat", which has the same data structure as the postdd.dat, the POSTDD code produces four output files: filename.out, filename.ou1, filename.ou2, and filename.ou3. A detailed user instruction on how to run this POSTDD code is summarized in Appendix A. In addition, two CDFs (based on 304lcl.ou1 and 304lcl.ou2)) and one PDF (based on 304lcl.ou3) are also plotted as Figures A-1 through A-3, respectively.

The filename.out file, which can be seen in Appendix A, summarizes the data input information. The filename.ou1 contains data which can be used to create CDFs at the user selected worst groups. The filename.ou2 contains data which can be used to create the CDFs at the user selected total number of cell failures. The filename.ou3 contains data which can be used to construct PDFs at the user selected simulation years. These files, except filename.out, are essentially two-column (X-Y) data files which can be easily incorporated with a graphical program to construct either CDF or PDF plots. Due to the simplicity of these three output files, no example files are illustrated.

5.6 MAJOR OUTPUT FILES SUMMARY

There are many output files created by the SMLEXC system. Most of these output files are intermediate results which are used to transfer data or to debug the program. The most important output

Table 5-6. sccsml.dat example

```
***** SCCEX FAILURE, WETTING, LEACHATE TIME *****   ! vector one results
 36 ! 36 calculation cells = 256 cells (symmetry)
 1 ! counter (first calculation cell)
 4 ! weight factor (similar waste packages)
  1000.000  0 ! failure time, type (0: no failure, 1: buckling, 2:  fracture, 3: yield, 4: corrosion)
  1000.000  0 ! wetting time, type (0: no wet, 1: wet)
  1000.000  0 ! leachate time, type (0: no leachate, 1: leachate)
 2
 8
  1000.000  0
  1000.000  0
  1000.000  0

  ! Data from 3-34

 35
  8
   1000.000  0
   1000.000  0
   1000.000  0
 36
 4
   1000.000  0
   1000.000  0
   1000.000  0
```

files are the end products created by the POSTDD code: filename.ou1, filename.ou2, and filename.ou3, where "filename" represents the input file name. These files can be used to construct CDFs as well as PDFs at the user-selected years, worst groups, and total number of cell failures. A demonstration example using the Alternative Case 2 (Cragnolino et al., 1994) data is shown in Appendix A. Two CDFs and one PDF, which are based on the Alternative Case 2 defined in the CNWRA report entitled "Substantially Complete Containment"—Example Analysis of a Reference Container, are also shown in Appendix A. The CDFs or PDFs can then be used to compare with the "substantially complete containment" aspects of 10 CFR Part 60.

6 SUMMARY AND CONCLUSION

The SMLEXC computer code has been developed to provide a computational framework using a probabilistic approach to construct CDFs and PDFs for failures of multiple waste packages contained in a simulated square repository. The CDFs and PDFs can be used to evaluate containment by a waste package and to identify those areas requiring more detailed analysis. As discussed in the report entitled "Substantially Complete Containment"—Example Analysis of a Reference Container (Cragnolino et al., 1994), both deterministic and probabilistic analyses were performed to provide valuable insight into the nature of the interdependence between the thermal, environmental, corrosion, and mechanical models, as well as the influence and importance of the various parameters associated with each model.

To implement the SMLEXC code, many simplifying assumptions and approximations were made to provide a quantitative representation of the actual conditions that may be presented in an unsaturated repository site. For deterministic analyses, for example, a simplified thermal model was adopted in the SCCEX code to facilitate parametric calculations. This simplified thermal model has limitations arising from the choice of outer boundary radius on the calculated value of temperature difference, and the fact that the temperature difference was assumed to be independent of the spacing between containers. For probabilistic analyses, ideally, calculations should be performed for each individual waste package and should exercise a significantly larger number of LHS simulations than the number used here. However, the computer time for such analyses is currently prohibitive. Other limitations are discussed in the report by Cragnolino et al. (1994).

The present results can be improved by using more advanced probabilistic analysis methods. As proposed in the report by Cragnolino et al. (1994), a two-step (probabilistic global/local) calculation process, which provides information for individual or a small number of packages, appears attractive. However, this approach may not work if many cells with a shorter life tend to fail at the same time, such that the selection of a worst cell in the first level analysis becomes impractical. To evaluate the requirement of substantially complete containment, it may be necessary to determine the failure time of individual waste packages and also determine the probability that only a small number of the packages fail before the design life. Therefore, it would be necessary to adopt numerically efficient methodologies that can represent the failure probabilities for single and multiple waste packages.

The present results can also be affected greatly by the random variables selected and their distribution functions. The current distribution functions and ranges of these random variables were selected from the available literature and information generated at the CNWRA. In the future, new data and additional research are needed to select the most appropriate distribution functions.

In summary, the SMLEXC methodology proposed appears to be a robust approach for evaluating waste package lifetimes in a geologic repository. Certainly further improvements are possible in the future, but the current SMLEXC framework has provided an initial step for evaluating the "substantially complete containment" aspects of 10 CFR Part 60.

7 REFERENCES

Cragnolino, G.A., N. Sridhar, J.C. Walton, R. Janetzke, T. Torng, J. Wu., and P.K. Nair. 1994. *Substantially Complete Containment—Example Analysis of a Reference Container*. CNWRA 94-003, San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Iman, R.L., and M.J. Shortencarrier. 1984. *A Fortran 77 Program and User's Guide for the Generation of Latin Hypercube Samples for Use with Computer Models*. SAND83-2365, NUREG/CR-3624. Albuquerque, NM: Sandia National Laboratories.

Janetzke, R.W., and B. Sagar. 1991a. *preFOR: A Pre-Processor for Fortran Files, User's Manual*. CNWRA 91-003. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Janetzke, R.W., and B. Sagar. 1991b. *RDFREE: A Fortran Utility for Format Free Input, User's Manual*. CNWRA 91-005. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Nuclear Regulatory Commission. 1990. *Disposal of High-Level Radioactive Wastes in Geologic Repositories*. Code of Federal Regulations, Part 60, Chapter 1, Title 10, "Energy." Washington, DC: Nuclear Regulatory Commission.

Nuclear Waste Policy Act. 1982. Public Law 97-425, 96 Stat 2201, 42USC 10101, 1983. Washington, DC.

Sridhar, N., J.C. Walton, G.A. Cragnolino, and P.K. Nair. 1993. *Engineered Barrier System Performance Assessment Codes (EBSPAC) Progress Report October 1, 1992 through September 25, 1993*. CNWRA 93-021. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

Torng, T.Y., P.C. Lichtner, and J.C. Walton. 1994. *SCCEX: A Computer Code For Substantially Complete Containment Example Analysis of a Reference Container, User's Manual*. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

TECPLOT. 1990. *TECPLOT User's Manual, Version 4*. Interactive Plotting for Scientists and Engineers. Bellevue, WA: Amtec Engineering, Inc.

U.S. Environmental Protection Agency. 1985. *Environmental Radiation Protection Standards for Management and Disposal of Spent-Nuclear Fuel, High-Level and Transuranic Radioactive Wastes: Final Rules*. Title 40, Code of Federal Regulations, Part 191. Federal Register. Washington, DC: U.S. Environmental Protection Agency. 20: 38065.

Wu, Y-T., A.G. Journel, L.R. Abramson, and P.K. Nair. 1991. *Uncertainty Evaluation Methods for Waste Package Performance Assessment*. NUREG/CR-5639. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.

APPENDIX A

POSTDD CODE USER INSTRUCTION

POSTDD CODE USER INSTRUCTION

The POSTDD code was developed to postprocess the system analysis data file, postdd.dat. This code is designed to fully utilize this information by constructing different cumulative distribution functions (CDFs) and probability density functions (PDFs) at a user-selected total number of cell failures or times to failure. These CDFs and PDFs can be easily used to compare the impact of different material and environmental parameters, to access the potential risk of the system at selected years, and to determine the necessary changes in design materials at selected reliability or probability-of-failure goals.

The POSTDD program is interactive, making it very convenient to use; however, the user must understand how to input the data correctly. The first step is to move the postdd.dat file to the subdirectory where the POSTDD executable resides. To illustrate the idea of how to use the POSTDD code, an example, which uses the postdd.dat results for the Alternative Case 2 (Cragnolino et al., 1994), is listed in Table A-1. A total of four data files will be created with this run. The names of these files and their functions are listed in Table A-2. The CDFs using 304lcl.ou1 and 304lcl.ou2 and the PDF using 304lcl.ou3 are plotted in Figures A-1 through A-3, respectively. As shown in Figures A-1 and A-2, the unevenness of the curves can be attributed to the insufficient sampling (only 100 LHS vectors used).

Table A-1. POSTDD code execution example

```
post.e ! to start the POST code
*** PROBABILISTIC OUTPUT POSTPROCESSOR ***  ! title
*** ENTER TOTAL NO. OF VECTORS USED =
100 ! important data in postdd.dat
*** ENTER TOTAL NO. OF GROUPS USED =
36 ! important data in postdd.dat
*** ENTER TOTAL NO. OF CELLS USED =
256 ! important data in postdd.dat
*** ENTER TOTAL SIMULATION TIME USED =
1000 ! important data in postdd.dat
*** ENTER TOTAL NO. OF GROUPS (ACTIVE CELLS) IN EACH VECTOR =
36  ! important data in postdd.dat
*** PROBABILISTIC DISTRIBUTION FOR THE WORST GROUPS ! based on mean  value
*** SELECT TOTAL NO. OF GROUPS TO BE CONSTRUCTED =
1
ENTER THE WORST GROUP NO. FOR DISTRIBUTION  1
1
*** PROB. DIST. AT USER-SELECTED TOTAL NO. OF CELL FAILURES
*** SELECT TOTAL NO. OF DISTRIBUTIONS TO BE CONSTRUCTED =
3
ENTER TOTAL NO. OF CELL FAILURES FOR DISTRIBUTION  1
1
ENTER TOTAL NO. OF CELL FAILURES FOR DISTRIBUTION  2
64
ENTER TOTAL NO. OF CELL FAILURES FOR DISTRIBUTION  3
256
```

Table A-1. POSTDD code execution example (cont'd)

```
*** PROBABILISTIC DISTRIBUTION AT USER-SELECTED YEARS
*** SELECT TOTAL NO. OF DISTRIBUTIONS TO BE CONSTRUCTED =
3
ENTER SELECTED YEAR FOR DISTRIBUTION  1
300
ENTER SELECTED YEAR FOR DISTRIBUTION  2
650
ENTER SELECTED YEAR FOR DISTRIBUTION  3
1000
*** ENTER DATA INPUT FILE NAME
304lcl.dat ! can be any other names (filename).
```

Table A-2. POSTDD code output files summary

Name	Function
"filename".out (e.g., 304lcl.out)	Data input echo (see Table A-1)
"filename".ou1 (e.g., 304lcl.ou1)	Data used for creating the CDF plots for the user selected worst groups
"filename".ou2 (e.g., 304lcl.ou2)	Data used for creating the CDF plots for user selected different total number of cell failures
"filename".ou3 (e.g., 304lcl.ou3)	Data used for creating the PDF plots at user selected years

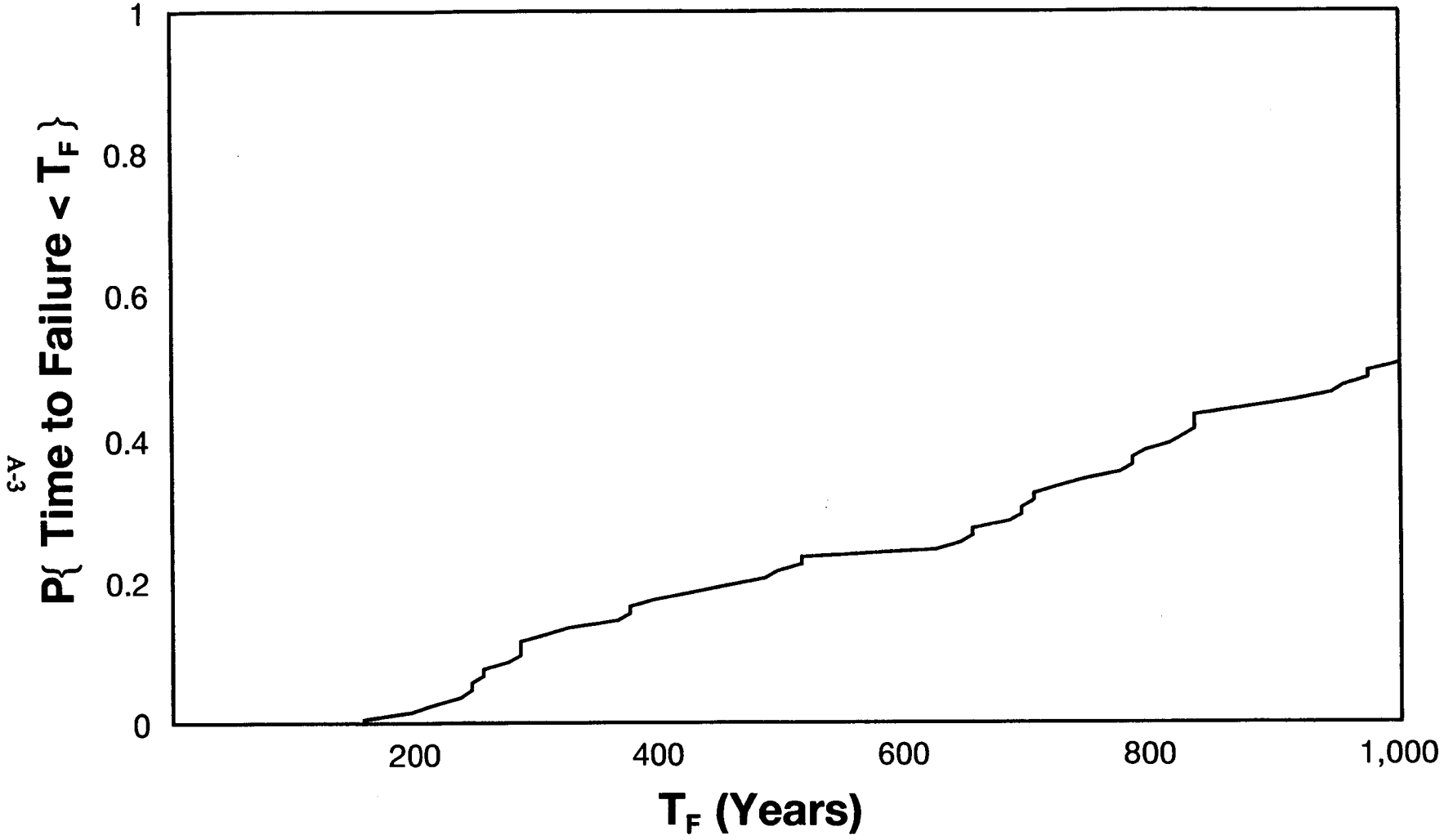


Figure A-1. Time-to-failure CDF for the worst group of cells corresponding to Alt-2 case (type 304L SS in controlled chloride environment) (Cragnolino et al., 1994)

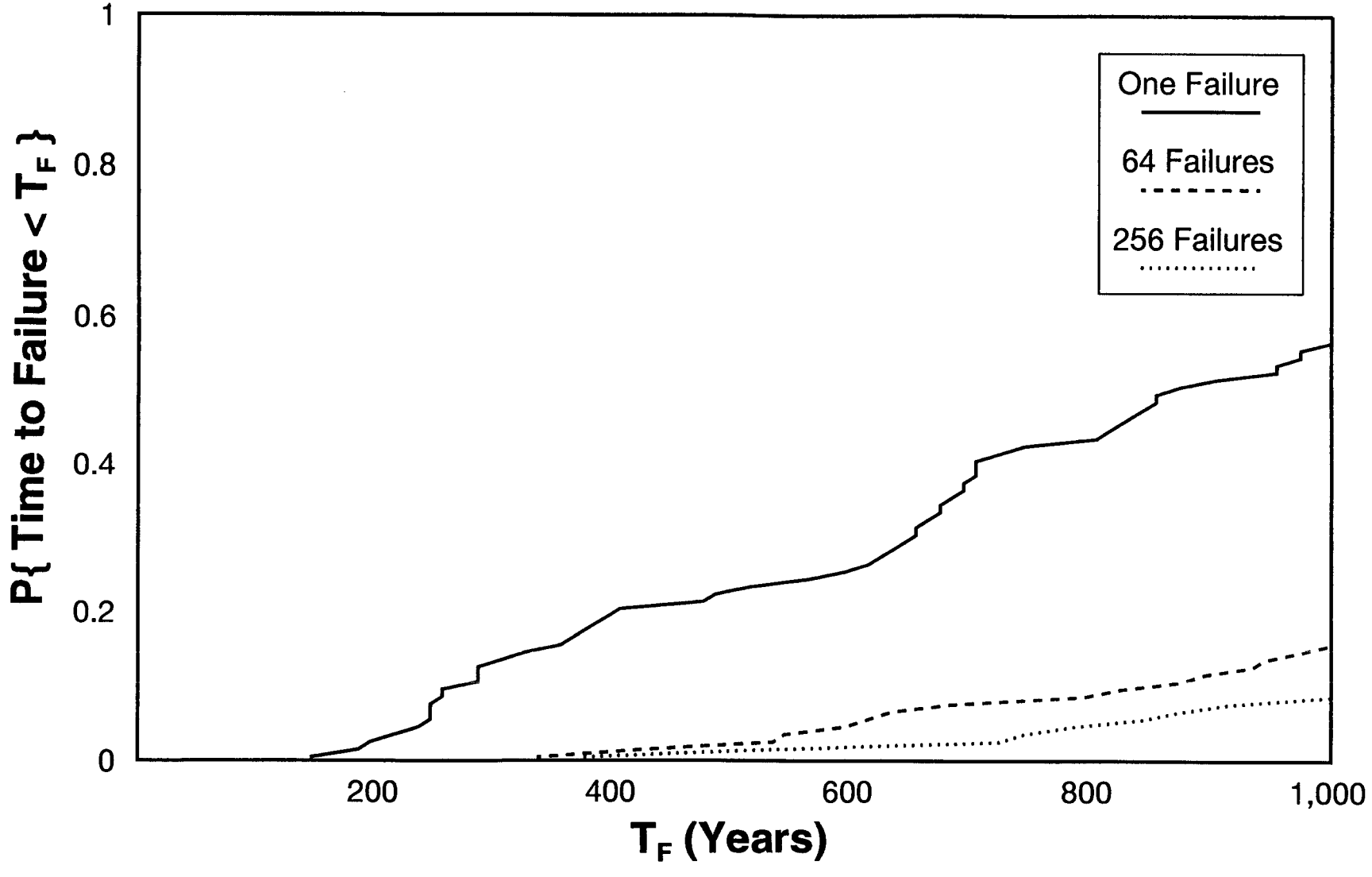


Figure A-2. CDF for selected cumulative number of cell failures corresponding to Alt-2 case (Cragnolino et al., 1994)

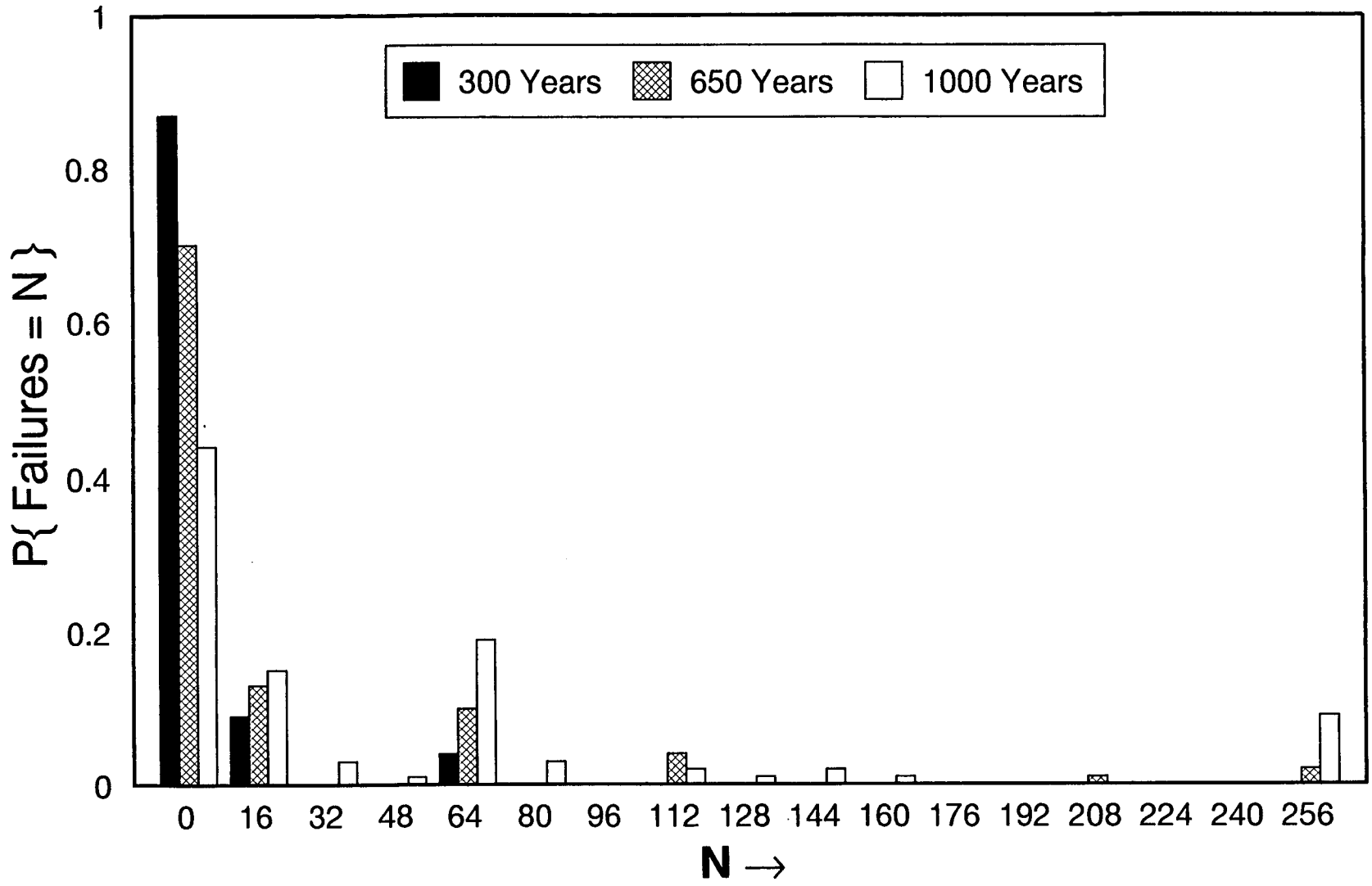


Figure A-3. PDF of the number of cell failures at selected times corresponding to Alt-2 case (Cragnolino et al., 1994)

APPENDIX B

ERROR MESSAGES

ERROR MESSAGES

'SMLEXC: Number of scenarios exceeds MAXSCN.'
'LHSEXC: LHS input file does not exist.'
'SCNINP: Unrecognized data for keyword ANALYS.'
'SCNINP: Unrecognized data for keyword ANALYS.'
'SCNINP: SCENAR keyword requires at least one modifier.'
'SCNINP: Unrecognized data for keyword SCENARIO.'
'SCNINP: Unrecognized data for keyword NAME.'
'SCNINP: Unrecognized data for keyword SCENARIO OUTPUT.'
'SCNINP: Unrecognized data for keyword WASTE.'
'SCNINP: Unrecognized data for keyword PROBAB.'
'SCNINP: Unrecognized data for keyword START.'
'SCNINP: Unrecognized data for keyword STOP.'
'SCNINP: Unrecognized data for keyword CANIST.'
'SCNINP: Unrecognized data for keyword CANLEN.'
'SCNINP: Unrecognized data for keyword CANRAD.'
'SCNINP: Unrecognized data for keyword CAN_WA.'
'SCNINP: Unrecognized data for keyword SCENARIO LHS.'
'SCNINP: Number of vectors exceeds MAXVEC.'
'SCNINP: Unrecognized data for keyword SCENARIO VECTOR.'
'SCNINP: Unrecognized data for keyword SCENARIO INVENT.'
'SCNINP: No other modifiers are permitted when the END modifier is used.' 'SCNINP: Number of consequence codes exceeds MAXMDL processing SCCEXP.' 'SCNINP: Unrecognized data for keyword SCCEXP INPUT.'
'SCNINP: Unrecognized data for keyword SCCEXP OUTPUT.'
'SCNINP: Unrecognized data for keyword SCCEXP LHSMAP.'
'SCNINP: Unrecognized data for keyword SCCEXP STOP.'
'SCNINP: Number of consequence codes exceeds MAXMDL processing SCCEX.' 'SCNINP: Number of consequence codes exceeds MAXMDL processing STRENG.' 'SCNINP: Unrecognized data for keyword LHS VECTOR.'
'SCNINP: Unrecognized data for keyword LHS INPUT.'
'SCNINP: Unrecognized data for keyword LHS OUTPUT.'
'SCPEXC: SCCEXP input file does not exist.'
'SOTEXC: LHS map file does not exist.'
'SCCEXC: SCCEX input file does not exist.'
'Requested READ file was not found.'
'NEW file creation blocked by existing file.'
'File status request is not valid.'