
Quality Assurance and Verification of the MACCS Code, Version 1.5

Manuscript Completed: January 1990
Date Published: February 1990

Prepared by
C. A. Dobbe, E. R. Carlson, N. H. Marshall,
E. S. Marwil, J. E. Tolli

DISCLAIMER

Idaho National Engineering Laboratory
Managed by the U.S. Department of Energy

EG&G Idaho, Inc.
P.O. Box 1625
Idaho Falls, ID 83415

Prepared for
Division of Systems Research
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555
NRC FIN A6896
Under DOE Contract No. DE-AC07-761D01570

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED
pe

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ABSTRACT

An independent quality assurance (QA) and verification of Version 1.5 of the MELCOR Accident Consequence Code System (MACCS) was performed. The QA and verification involved examination of the code and associated documentation for consistent and correct implementation of the models in an error-free FORTRAN computer code. The QA and verification was not intended to determine either the adequacy or appropriateness of the models that are used in MACCS 1.5. The reviews uncovered errors which were fixed by the SNL MACCS code development staff prior to the release of MACCS 1.5. Some difficulties related to documentation improvement and code restructuring are also presented. The QA and verification process concluded that Version 1.5 of the MACCS code, within the scope and limitations of the models implemented in the code, is essentially error free and ready for widespread use.

EXECUTIVE SUMMARY

The development of the MELCOR Accident Consequence Code System (MACCS) by Sandia National Laboratories (SNL) has been funded by the US Nuclear Regulatory Commission's Office of Nuclear Regulatory Research for the past several years. MACCS models the offsite consequences of hypothetical accidental atmospheric releases of radioactive material. MACCS calculational features include atmospheric dispersion and transport, deposition on ground and water, environmental transfer of radioactive material, mitigative actions, dosimetry, health effects and economic consequences. Earlier versions of the MACCS computer code were used to calculate the consequences for the first draft NUREG-1150, *Reactor Risk Reference Document*, published in February 1987. Criticisms were leveled at these versions of MACCS through comments on draft NUREG-1150 reflecting on the code's accuracy and adequacy. The current code version, MACCS Version 1.5 (MACCS 1.5), has corrected most of these deficiencies and incorporated several new features including a new meteorological bin sampling scheme, provisions for accommodating wind shifts between release phases, multiple run capability, etc. MACCS 1.5 was used for the second draft NUREG-1150, *Severe Accident Risks: An Assessment of Five U.S. Nuclear Power Plants*, completed in 1989.

The completion of the MACCS 1.5 computer code development and appropriate documentation (User's Guide, Model Description and Programmer's Reference Manual) would allow for an independent quality assurance (QA) and verification of MACCS 1.5 before its widespread use by the NRC staff in regulatory activities. In particular, significant QA and verification needed to be performed before calculations were performed for the second draft NUREG-1150 completed in 1989. Because the MACCS 1.5 documentation was not yet complete prior to the calculations for the second draft NUREG-1150, other sources of information were utilized for model descriptions in lieu of the Model Description. These sources included walk-through of the MACCS FORTRAN with the SNL code developers, model descriptions from earlier MACCS code version manuals and original reference material for the various code models. The QA and verification then identified errors and deficiencies in the MACCS code relative to these various model descriptions. These were then provided to SNL for incorporation of corrections into MACCS 1.5 and the resulting code would then be released for use by the NRC staff.

The objective of this task was to perform an independent QA and verification of the MACCS 1.5 computer code. The QA and verification process was defined for this task as verifying that the models are implemented as described into a consistent, error-free FORTRAN computer code. The QA and verification process involved: examination of the individual code modules for internal and interfacing consistencies; evaluation of the FORTRAN statements, algorithms, statistical techniques and input data requirements to ensure accurate and consistent implementation; verification of the output capabilities; and the code's overall evaluation per state-of-the-art coding standards for large scientific computer programs. The documentation provided with the MACCS 1.5 code was also reviewed for accuracy and completeness. The reviews were not intended to determine either the adequacy or appropriateness of the models that are used in the MACCS 1.5 code. These determinations will need to be made through the ongoing process of benchmarking analyses with MACCS 1.5 and comparisons of the various MACCS 1.5 models and calculational schemes to other consequence codes.

The line-by-line verification of MACCS 1.5 was performed as a series of parallel reviews of the distinct code program modules. The code program modules included ATMOS, which calculates atmospheric dispersion, EARLY, which calculates the early health effects and mitigative actions, and CHRONC, which calculates the long-term health and economic effects. The balance of the code, comprised primarily of the input and output routines, was reviewed separately. The overall code evaluation was a separate task that concentrated on the FORTRAN to ascertain code logic paths, module timings and adherence of the coding to ANSI standards. The MACCS 1.5 draft code documentation provided by SNL was reviewed for completeness and accuracy as part of the QA and verification effort. The documentation is composed of three volumes. Volume I, the User's Guide, describes the input data requirements of MACCS 1.5 and provides directions for using the code. Volume II, the Model Description, describes the underlying models that are implemented in MACCS 1.5. Volume III, the Programmer's Reference Manual, describes the MACCS 1.5 code structure and database management. The documentation of a parallel effort for comparative and comprehensive review of the chronic exposure pathway models related to the QA and verification of MACCS 1.5 was also reviewed and summarized.

The latest iteration of QA and verification of the MACCS 1.5 computer code shows that there are no obvious uncorrected errors in the code and that it is ready for release. The earlier reviews of the various code modules and documentation uncovered errors which could have affected MACCS 1.5 computational results. These have been fixed by the SNL code developers. Other items that have not as yet been addressed by SNL affect code readability, code maintainability and understanding of model implementation in MACCS 1.5. The primary items in this category include parameterization of common blocks and other program fragments, use of an INCLUDE facility to propagate changes through the code, a rewrite of the ATMOUT subroutine, a rewrite of the meteorological binning routines, and providing both the source codes and documentations for the dose conversion factors and the maximum permissible

ground concentrations preprocessor programs.

The draft documentation provided with MACCS 1.5 was reviewed and the later draft User's Guide was found to accurately represent the code. The Model Description and Programmer's Reference Manual contained numerous errors that are expected to be corrected in the final documentation. Requested enhancements to the write-ups in the Model Description should help improve readability and understanding of the models implemented in MACCS 1.5. Additional documentation that currently exists only in draft form, if at all, relative to the MACCS input used for the NUREG-1150 revisions (NUREG/CR-4551) and the chronic exposure pathway models (NUREG/CR-4814) would provide valuable information for the preparation of MACCS 1.5 input and therefore should be completed and issued.

ACKNOWLEDGMENTS

The authors would like to thank Sarbes Acharya of the U. S. Nuclear Regulatory Commission for his contributions in preparing this report. The Sandia National Laboratory MACCS code development staff, particularly David Chanin, provided invaluable assistance in all aspects of the work described in this report.

CONTENTS

ABSTRACT	iii
EXECUTIVE SUMMARY	iv
ACKNOWLEDGMENTS	vi
1. INTRODUCTION	1
2. MACCS VERSION 1.5 CODING QA AND VERIFICATION	2
2.1 Static and Dynamic Analysis of the MACCS 1.5 FORTRAN Code	2
2.2 Atmospheric Transport, Dispersion, and Deposition Routines	12
2.3 Early Consequence Routines	15
2.4 Chronic Consequence Routines	20
2.5 Meteorological Data Routines	23
2.6 Input Processing and Miscellaneous Routines	28
3. MACCS 1.5 DOCUMENTATION REVIEW	35
3.1 User's Guide	35
3.2 Model Description	37
3.3 Programmer's Reference Manual	39
4. SUPPORTING REVIEW AND ANALYSES	42
5. CONCLUSIONS	43
REFERENCES	44
DISTRIBUTION	45

TABLES

1. Statement type summary from the FORTRAN 77 analyzer for MACCS 1.5.0	4
2. Unreferenced common blocks in MACCS 1.5.0	5
3. The twenty MACCS 1.5.0 program units with the most FORTRAN statements	7
4. The twenty MACCS 1.5.0 program units with the highest difficulty metric	8
5. The twenty MACCS 1.5.0 program units with the greatest control flow complexity	8
6. Timing profile of the MACCS 1.5.0 sample problem calculation for the ten program units requiring the most time	10

7.	Timing profile of the MACCS 1.5.0 sample problem calculation for the ten program units called the most times	10
8.	Subroutines and functions reviewed during the ATMOS program module QA and verification	12
9.	Subroutines and functions reviewed during the EARLY program module QA and verification	16
10.	Subroutines and functions reviewed during the EARLY program module QA and verification	21
11.	Errors detected in the draft MACCS 1.5 User's Guide during the CHRONC program module review	22

QUALITY ASSURANCE AND VERIFICATION OF THE MACCS CODE, VERSION 1.5

1. INTRODUCTION

The development of the MELCOR Accident Consequence Code System (MACCS)^a by Sandia National Laboratories (SNL) has been funded by the US Nuclear Regulatory Commission's Office of Nuclear Regulatory Research for the past several years. MACCS models the offsite consequences of hypothetical accidental atmospheric releases of radioactive material. MACCS calculational features include atmospheric dispersion and transport, deposition on ground and water, environmental transfer of radioactive material, mitigative actions, dosimetry, health effects, and economic consequences. Earlier versions of the MACCS computer code were used to calculate the consequences for the first draft of NUREG-1150, published in February 1987.¹ Criticisms were leveled at these versions of MACCS through comments on the first draft of NUREG-1150 reflecting on the code's accuracy and adequacy. The current code version, MACCS Version 1.5 (MACCS 1.5), has corrected most of these deficiencies and incorporated several new features including a new meteorological bin sampling scheme, provisions for accommodating wind shifts between release phases, multiple run capability, etc. MACCS 1.5 was used for the second draft NUREG-1150 completed in 1989.² The MACCS 1.5 calculations that were performed for the second draft of NUREG-1150 are referred to in this report as the NUREG-1150 revisions.

The completion of the MACCS 1.5 computer code development and appropriate documentation (User's Guide, Model Description and Programmer's Reference Manual) would allow for an independent quality assurance (QA) and verification of MACCS 1.5 before its widespread use by the NRC staff in regulatory activities. In particular, significant QA and verification needed to be performed before calculations were performed for the NUREG-1150 revisions completed in 1989. Since the MACCS 1.5 documentation was not yet complete prior to the calculations for the NUREG-1150 revisions, other sources of information were utilized for model descriptions in lieu of the Model Description. These sources included walkthroughs of the MACCS FORTRAN with the SNL code developers, model descriptions from earlier MACCS

code version manuals and original reference material for the various code models. The QA and verification then identified errors and deficiencies in the MACCS code relative to these various model descriptions. These were then provided to SNL for correction of MACCS 1.5 and the resulting code would then be released for use by the NRC staff.

The objective of this task was to perform an independent QA and verification of MACCS 1.5. The QA and verification process was defined for this task as verifying that the models are implemented as described into a consistent, essentially error-free FORTRAN computer code. The QA and verification performed involved: examination of the individual code modules for internal and interfacing consistencies; evaluation of the FORTRAN statements, algorithms, statistical techniques and input data requirements to assure accurate and consistent implementation; verification of the output capabilities; and the code's overall evaluation for quality per state-of-the-art coding standards for large scientific computer programs. The documentation provided with the MACCS 1.5 code was also reviewed for accuracy and completeness. The reviews were not intended to determine either the adequacy or appropriateness of the models that are used in the MACCS 1.5 code. These determinations will need to be made through the process of benchmarking analyses with MACCS 1.5 and comparisons of the various MACCS 1.5 models and calculational schemes to other consequence codes.

The following report documents the findings of these reviews and the ultimate disposition of any identified errors or deficiencies. The code modifications made by SNL as required to address these errors and deficiencies are separated into two groups. Group 1 represents changes incorporated into the code version used for the NUREG-1150 revisions and group 2 represents changes made to the released version of MACCS Version 1.5. What remains are the suggested changes proposed during the reviews that are not yet incorporated into MACCS 1.5. The MACCS Version 1.5 code QA and verification is discussed on a module-by-module basis in Section 2. The results of the MACCS Version 1.5 documentation review is provided in Section 3. Section 4 provides a brief overview of other reported work that was related to the QA and verification of MACCS 1.5.

a. D. I. Chanin et al., *MELCOR Accident Consequence Code System (MACCS)*, NUREG/CR-4691, SAN86-1562, Sandia National Laboratories, to be published.

2. MACCS VERSION 1.5 CODING QA AND VERIFICATION

The following documents the results of the line-by-line QA and verification process for the MACCS 1.5 computer code. The versions used for this task were the August 15, 1988 draft of MACCS 1.5 (Version 1.5.0), the July 22, 1988 draft and May 11, 1989 draft of the MACCS 1.5 User's Guide, and the May 27, 1987 draft of the MACCS Version 1.4 Reference Manual. The cited documents were draft versions of the final documentation prepared for the released version of MACCS 1.5. Additional documentation utilized for the line-by-line reviews of the various MACCS program modules are described in the following discussions. Walkthroughs of the coding with the SNL MACCS code development staff were required for the main program modules to obtain appropriate model discussions and discuss basic program logic and inter- and intra-module information flow.

The line-by-line verification of MACCS 1.5 was performed as a series of parallel reviews of the distinct code program modules. The code program modules included ATMOS, which calculates atmospheric dispersion, EARLY, which calculates the early health effects and mitigative actions, and CHRONC, which calculates the long term health and economic effects. The balance of the code, comprised primarily of the input and output routines, was reviewed separately. Due to a lack of complete documentation for MACCS 1.5, each of these tasks required an initial coding walkthrough and subsequent consultations with the SNL MACCS code development staff. The overall code evaluation was a separate task that concentrated on the FORTRAN to ascertain code logic paths, module timings and adherence of the coding to ANSI standards. Close coordination between INEL, SNL and NRC was maintained throughout these reviews with identified modifications and/or corrections transmitted as they were discovered for correction of MACCS 1.5.

The results of these individual module reviews are documented in the following sections. These sections summarize the findings of the reviews with emphasis given to items that reflect on the overall adequacy of MACCS 1.5. The code errors and/or deficiencies discovered are explicitly described in order to provide a complete record of the QA and verification process. MACCS 1.5 is the first official release of the code and prior versions may or may not have the described errors. As stated earlier, these errors and/or deficiencies are identified relative to when or if they were incorporated into MACCS 1.5. This was done to provide some verification for the version of MACCS 1.5 used for the

NUREG-1150 revisions, Version 1.5.5. Adequacy of the documentation refers here to the documentation available for this review. This information is provided since the Reference Manual used in the QA and verification was unofficially distributed with MACCS Version 1.4. A more detailed discussion of the MACCS 1.5 documentation is provided in Section 3 of this report.

2.1 Static and Dynamic Analysis of the MACCS 1.5 FORTRAN Code

The purpose of this work is to report results of global static and dynamic tests which were run on MACCS 1.5 and to record some recommendations as a result of this analysis. The following information refers to the initial version of MACCS 1.5, designated Version 1.5.0, and hereafter referred to as MACCS 1.5.0. Analysis of the MACCS code was done using available software tools which include the following:

- FTREF³—This tool gives information relative to common block usage. It will detect many instances where variables in common are used but never defined. FTREF can also be used to create common block vs. subroutine cross reference tables and called routine vs. calling routine cross reference tables.
- TOOLPACK⁴—Used for static analysis of selected subroutines from MACCS 1.5.0.
- FOLSM^a—FOLSM was used to determine the Halstead and the McCabe complexity metrics for MACCS 1.5.0.
- CFT77⁵—CRAY FORTRAN compiler. Three options in particular were used. These options are:

The option which flags non-ANSI standard FORTRAN features

The option which detects array subscripts that are out of bounds

a. The FOLSM program was obtained from Loren Meissner of the University of San Francisco in 1988.

The option which creates a timing profile of the code under examination.

- FORTRAN 77 Analyzer⁶ — An analysis tool used to give the occurrences distribution of the kinds of FORTRAN statements which are in MACCS 1.5.0.

In general, the analyses can be broken into two parts: static analysis, which is performed without executing the program, and dynamic analysis, which analyzes certain aspects of and therefore requires program execution. For the dynamic analysis reported here, the sample problem transmitted with MACCS 1.5.0 was used.

The static and dynamic analyses of MACCS 1.5.0 produced evidence of carefully structured and tested FORTRAN coding. Tests which often disclose program errors in production codes did not identify any errors in MACCS 1.5.0. This, of course, does not necessarily imply that there are no errors in MACCS 1.5.0, only that none were detected. The static analysis indicates that statements which tend to obscure program readability and reduce program maintainability were missing or used sparingly in MACCS 1.5.0. On the other hand, the block IF construct (a highly structured construct) was often used. This indicates a conscious effort on the part of the code developers to structure the FORTRAN code. The program design and modular structure were not analyzed, but there was some evidence that perhaps it should be.

The static analysis was performed with the FORTRAN source code of MACCS 1.5.0. Because the MACCS code was not exercised in the static analysis, the analysis was limited to those problems which could be detected without executing the code. The static analysis of MACCS 1.5.0 was performed to determine statement distributions, program clutter, ANSI standard conformance, program complexity measures, and common block usage. The following paragraphs present the results of this analysis.

The first item evaluated was the statement distribution within the MACCS code. MACCS 1.5.0 is a FORTRAN program which contains:

- 29,614 source lines of code containing 3,945 blank lines
- 14,075 comment lines
- 10,047 FORTRAN statements
- 1,547 continuation lines.

MACCS 1.5.0 also contains 194 common blocks and 206 program units. A program unit is defined to be a main program, a subroutine, a function, or a block data subprogram. The distribution of the FORTRAN statements in MACCS 1.5.0 is presented in Table 1.

It was observed from the statement distribution that over ten percent of the FORTRAN statements are COMMON statements (global data). This seems a bit high from a software engineering standpoint, however, MACCS 1.5.0 does contain a considerable number of common blocks which are never referenced. These common blocks are listed in Table 2 along with the program units which contain but never reference them. Removal of these unreferenced common blocks would reduce this statistic appreciably. SNL removed these unreferenced COMMON statements from MACCS 1.5 after the NUREG-1150 revisions.

Statements which lead to so called "spaghetti code," such as the GO TO statement, are low. Statements which inhibit program readability and maintenance (such as EQUIVALENCE statements) are also very low. This indicates a conscientious effort to avoid these particular statements. On the other hand, the percentage of the block IF (6.7%) and the logical IF (4.8%) are quite high. Comparing these percentages with those for the GO TO statement shows a very favorable balance. This indicates considerable effort on the part of the MACCS authors to use structured programming techniques.

MACCS uses several external files yet there are no FORTRAN OPEN statements in MACCS 1.5.0. Since OPEN statements provide a standard way of connecting files and logical units, this was a little surprising. MACCS 1.5 reads from unit numbers and does not use FORTRAN OPEN to connect files and logical units. SNL states that was intentional since the FORTRAN standard does not have a specification for file names or defaults (TAPExx on CDC, FORT.xx on CRAY, etc.). To avoid this system dependency, SNL included a VAX/VMS COMMAND file with the transmittal tape to show the relationship between input files and units. However, it was discovered that, because of the way files were read by MACCS 1.5, failure to have a required external file attached could result in infinite looping during execution. The released version of MACCS 1.5 contains explicit OPEN and CLOSE statements for all external input, data, and output files, put into MACCS 1.5 after the NUREG-1150 revisions.

MACCS 1.5.0 does contain some of what is often referred to as "program clutter." Program clutter is made up of FORTRAN statements or parts of statements which do not contribute to the FORTRAN program. These are entities that are not used in any way by the program. While the impact of clutter is usually not seri-

ous, it does have a negative impact on the overall program from the standpoint of program size and readability. Kinds of program clutter which were

identified in MACCS 1.5.0 include unreferenced common blocks, unreferenced program units, unused external references, and unreferenced statement labels.

Table 1. Statement type summary from the FORTRAN 77 analyzer for MACCS 1.5.0

<u>Statement Type</u>	<u>Number of Occurrences</u>	<u>%</u>	<u>Statement Type</u>	<u>Number of Occurrences</u>	<u>%</u>
ASSIGN	0	0.0	GO TO	121	1.2
BACKSPACE	0	0.0	(assigned)	0	0.0
BLOCK DATA	0	0.0	(computed)	0	0.0
CALL	532	5.3	(unconditional)	121	1.2
CHARACTER	270	2.7	IF	1162	11.6
CLOSE	1	0.0	(arithmetic)	1	0.0
COMMON	1011	10.1	(block)	677	6.7
COMPLEX	0	0.0	(logical)	484	4.8
CONTINUE	328	3.3	IMPLICIT	0	0.0
DATA	72	0.7	INQUIRE	0	0.0
DIMENSION	80	0.8	INTEGER	17	0.2
DOUBLE PRECISION	0	0.0	INTRINSIC	43	0.4
DO	782	7.8	LOGICAL	148	1.5
ELSEIF	146	1.5	OPEN	0	0.0
ELSE	200	2.0	PARAMETER	9	0.1
ENDFILE	0	0.0	PAUSE	0	0.0
END IF	677	6.7	PRINT	0	0.0
END	206	2.1	PROGRAM	1	0.0
ENTRY	6	0.1	READ	66	0.7
EQUIVALENCE	5	0.0	REAL	14	0.1
FORMAT	255	2.5	REWIND	11	0.1
FUNCTION	35	0.3	SAVE	125	1.2
(character)	1	0.0	STOP	0	0.0
(double precision)	0	0.0	SUBROUTINE	170	1.7
(integer)	0	0.0	WRITE	509	5.1
(logical)	5	0.0	ASSIGNMENT statements	2600	25.9
(real)	0	0.0	STATEMENT functions	21	0.2
(untyped)	29	0.3	Unrecognized statements	0	0.0

Table 2. Unreferenced common blocks in MACCS 1.5.0

<u>Common Block</u>	<u>Program units contain- ing but not referencing common blocks</u>	<u>Common Block</u>	<u>Program units contain- ing but not referencing common blocks</u>
ACANCR	OUTPTI	KPRINT	WGCPLN, CSTEFF
ATMDAT	INEVAC, OPNERL	METOUT	ATMOUT
ATMOPT	INPREL, WBNDRY, PLMRIS	NETWDR	OPNERL, EMRGPH
ATNAM1	OUTCON	NUMRES	OUTCON
ATNAM2	OUTCON	NUMVAL	OUTCON, PRINT
CHNAME	OUTCON	NXMRES	OUTCON
CRPTIM	STTIMS, LTMACT	NXMVAL	OUTCON
EANAN1	OUTCON	ORGNAM	OUTPT3
EANAN2	OUTCON	OUTCOM	DO1CDF
EINJUR	OUTPT1	PHYCON	CAUGHT
ERLCST	CSTEFF	RESNAM	OUTCON
FRACLD	STPATH	REUSE2	READ2
FRCFRM	STPATH, CSTEFF	RXSNAM	OUTCON
GDFDEC	LTMACT	SITEDT	LTMACT
GLOBAL	PUTSTG, OUTCON, READ1	SRCTRM	OUTCON
GSWTHR	STPATH	SRZONE	OPNERL, INCDOS, EMOVE, INCREM, EMRGPH
HEADER	OUTCON	STOPME	PRINT
IHITIT	RELZON	STRTGY	OUTCON
INPRC3	CGET1, IGET1, LGET1, RGET1	THYDAT	OPNERL (note: this common block is never accessed)
ISOGRP	INPOPT, EPCALC		
ISONAM	INORGA, SGCPLN, WGCPLN	TIMES	INTRPH
		TIMIPP	EMRGPH
IUNIT	PRINT	WTFRAC	OUTCON
KKPRNT	CHRINP	WTNAME	PUTSTG, OUTCON

A common block may be referenced in any number of program units, but when it occurs in a program unit in which it is not referenced, it becomes program clutter. In fact, some authors, such as Glenford Myers, feel that even unreferenced variables within a common block constitute clutter. Concerning these he writes: "Hence, there are variables in the module's listing that are not referenced, a possible source of confusion. A larger problem is that the module has addressability to this unneeded data. The module might inadvertently alter this data, leading to an obscure error. A more frequent problem is that the programmer might attempt to take advantage of the presence of this additional data, thus reducing further the independence within the program."⁷ As discussed earlier, SNL removed these unreferenced COMMON statements listed in Table 2. They have not, as yet, removed unreferenced variables from common blocks within MACCS 1.5. SNL indicated that they would prefer to rework the common blocks within MACCS 1.5 to provide a more rational grouping of the variables. At that time, they would also remove any unreferenced variables.

In MACCS 1.5.0, there are three instances where the clutter occurs in the form of program units which are never referenced. These program units are COMPRS, CCDOUT, and POWR10. The routines COMPRS and POWR10 contain comments to the effect that they are no longer used. There was no justification given for their continued presence in the MACCS code. SNL removed CCDOUT and POWR10 from the code after the NUREG-1150 revisions. In the released version of MACCS 1.5, COMPRS is accessed by subroutine RESLT3 and remains in the code although it still contains the comment statement that the program unit is unused by MACCS 1.5.

Another minor form of clutter in MACCS 1.5.0 is when external program units are declared to be external, and then the program unit itself is never referenced. Four instances of this kind of clutter were identified in MACCS 1.5.0. These are external RGET1 in program unit INPM3, external ERRLOC in program unit INOUT2, external IGETN in program unit INOUT3, and external CRNRSK in program unit CHRINP. These external calls are still present in the released version of MACCS 1.5.

The final form of clutter which occurs in MACCS 1.5.0 is unreferenced statement labels. Again, these are not serious, however, in some instances they may inhibit program optimization. Some instances, such as label 40 in program unit MACCS, have been identified, however, there was not an extensive search to identify instances of this kind of clutter.

MACCS 1.5.0 was checked for non-ANSI standard FORTRAN usage using the CFT77 compiler on the CRAY X-MP/24. The only non-ANSI standard usage located were 31 instances where subroutine ABORTMC was called. Since this name exceeds the six character limit for FORTRAN names, these calls were flagged by the compiler as non-ANSI. This is not a serious error and currently there are several FORTRAN compilers (including CFT77) which accept the longer names. In fact, this particular routine name was put into the code at the INEL. MACCS 1.5 has a routine called ABORT which caused problems with an earlier version of the CFT77 compiler because the name was the same as a system routine on the CRAY X-MP/24. The name was changed to ABORTMC to alleviate this problem. The current CFT77 compiler will allow routine names to supersede system routines as allowed in the ANSI standard and the name ABORTMC was changed back to ABORT prior to the NUREG-1150 revisions.

FOLSM was used to compute three complexity measures or "metrics" as they are referred to in software engineering terms. The three used for examining MACCS 1.5.0 are the number for FORTRAN statements (excluding comment lines), Halstead's difficulty metric,⁸ and McCabe's cyclomatic complexity.⁹ The count of FORTRAN statements is considered by some to be rather inelegant but it still conveys meaningful information. Table 3 lists the 20 MACCS 1.5.0 program units with the most number of FORTRAN lines. Halstead's difficulty metric was developed by Maurice Halstead when he introduced his theory called Software Science, a series of software metrics utilized to analyze the number of operators and operands in a program. Table 4 lists the 20 MACCS 1.5.0 program units with the greatest difficulty metric. Tom McCabe's complexity metric is a frequently quoted metric based upon the number of flow paths in a program. Higher values of this metric indicate more logical flow paths through the program. Table 5 lists the 20 MACCS 1.5.0 program units with the highest control flow complexity. Each of the tables of metrics show the other two metrics for the program units listed for comparison purposes.

Complexity metrics give a relative indication of how difficult it may be to understand the relevant coding and, therefore, how difficult it may be to modify it. In general, the most complex program units will be the most likely to contain programming errors. They are also likely to be the most difficult program units to maintain. The complexity analysis was not used to identify code errors but to direct the review to those routines most likely to have errors.

As stated earlier, MACCS 1.5.0 contains 194 common blocks. The usage of entities in common was examined using FTREF and SCAN. These routines revealed the existence of unreferenced common blocks in program units discussed earlier and shown in Table 2. What is important to note is that these tests did not reveal any instance where an entity in a common block was not initialized, an error that exists in many production codes. Since these tools did not perform any type of a path analysis, one cannot state that all entities in common were initialized before they are first used, only that all entities in common are initialized somewhere in the code. The analysis also revealed that almost all entities in common have the same name and attributes in all of the program units in which the common block resides. This is consistent with good programming practices.

Dynamic analysis was performed on MACCS 1.5.0 using the sample problem provided with the code. As

the sample problem was based on a NUREG-1150 input deck it was felt to provide a fairly rigorous test of the code. The dynamic analysis was performed during the execution of the sample problem to reveal performance characteristics.

The first task performed was array bounds checking. The CFT77 compiler was used to perform this test to identify any time an array index goes outside of the prescribed array bounds. With this compiler option active, MACCS 1.5.0 executed the sample problem without a single instance of array bound violation being detected.

Another option of the CFT77 compiler is the ability to perform a timing analysis on the program unit level. This analysis was also performed on MACCS 1.5.0 while it executed the sample problem. The timing profile for MACCS 1.5.0 reported the time each program unit was executing, the number of times each program

Table 3. The twenty MACCS 1.5.0 program units with the most FORTRAN statements

<u>Routine Name</u>	<u>Fortran Statements</u>	<u>Difficulty Metric</u>	<u>Control Flow Complexity</u>
STPATH	345	178.55	112
ATMOUT	322	92.01	83
WBNMET	196	71.52	39
SDFINP	184	49.83	34
INCHRN	179	89.05	100
PRINT	169	65.13	36
EXCINP	149	58.00	35
INPREL	142	74.77	52
ESTAT	138	62.00	44
INPBEG	131	60.09	22
EDCINP	130	69.17	27
COPCHR	122	43.58	27
OPNERL	116	34.74	14
INOUT7	114	56.55	36
INPOPU	113	50.93	19
EVNETW	112	74.26	37
INOUT1	106	52.77	35
INOUT8	106	52.94	35
HEDEAR	105	41.84	26
EPCALC	104	55.35	28

Table 4. The twenty MACCS 1.5.0 program units with the highest difficulty metric

<u>Routine Name</u>	<u>Fortran Statements</u>	<u>Difficulty Metric</u>	<u>Control Flow Complexity</u>
STPATH	345	178.55	112
IMREAL	58	121.09	34
ATMOUT	322	92.01	83
INCHRN	179	89.05	100
RDISTB	71	75.72	39
INPREL	142	74.77	52
OXTPT8	82	74.57	42
EVNETW	112	74.26	37
WBNMET	196	71.52	39
EDCINP	130	69.17	27
QUANTL	55	66.35	28
OXTPT1	65	65.55	36
PRINT	169	65.13	36
INPISO	94	64.93	27
INPEMR	62	63.51	31
RDSTRG	94	62.62	24
INPM4	85	62.38	32
ESTAT	138	62.00	44
EMOVE	98	61.71	31
INPBEG	131	60.09	22

Table 5. The twenty MACCS 1.5.0 program units with the greatest control flow complexity

<u>Routine Name</u>	<u>Fortran Statements</u>	<u>Difficulty Metric</u>	<u>Control Flow Complexity</u>
STPATH	345	178.55	112
INCHRN	179	89.05	100
ATMOUT	322	92.01	83
INPREL	142	74.77	52
ESTAT	138	62.00	44
OXTPT8	82	74.57	42
RDISTB	71	75.72	39
WBNMET	196	71.52	39
EVNETW	112	74.26	37

Table 5. (continued)

<u>Routine Name</u>	<u>Fortran Statements</u>	<u>Difficulty Metric</u>	<u>Control Flow Complexity</u>
OXTPT1	65	65.55	36
PRINT	169	65.13	36
INOUT7	114	56.55	36
PUTSTG	99	48.17	36
EXCINP	149	58.00	35
LTACUM	92	57.35	35
INOUT8	106	52.94	35
INOUT1	106	52.77	35
IMREAL	58	121.09	34
SDFINP	184	49.83	34
OXPT11	67	52.53	33

unit was called, and the average time per call. Table 6 lists in descending order the 10 program units that used the most time. Table 7 lists in descending order the 10 program units that were called the most times. It is interesting to note that for the sample problem approximately two-thirds of the time was spent in two program units, EFFGET (32.1%) and OUTPT1 (31.1%). It is also interesting to note that EFFGET was invoked more than 14 million times during the sample run.

Timing profiles are often used to provide information used in optimizing computer programs. Once program units which use the significant amount of time are identified, the optimization effort can be concentrated in those program units. Because invoking another program unit entails a certain amount of overhead, an excessive number of calls can also indicate an area where some execution time savings may be realized. Reducing the number of calls may require program structure redesign, therefore, some analysis needs to be done relative to cost of program redesign compared to potential cost savings.

Based on the timing analysis, SNL decided to evaluate OUTPT1 and EFFGET. They discovered that these routines, which produce results for the EARLY program module, were used as a starting point for writing similar routines in the CHRONC program module. These routines, OXTPT1 and CASGET, only required 3% of the cpu time for the sample problem run. Although the amount of time required is directly related

to the number of output results requested from each module, comparing the timing results showed that the potential existed for a significant time savings. The dynamic coding analysis seems to indicate a substantial (30 - 50%) savings in cpu if the code was also optimized in EARLY. However, the CASGET routine is vectorizable on the INEL CRAY X-MP/24 whereas the same routine in EARLY is probably not. Thus the savings may not be as great as the dynamic analyses would tend to indicate but optimization was still thought advantageous to reduce the number of calls to EFFGET. SNL proposed revising OUTPT1 to request all early results for each spatial element in a single call. This would reduce the number of calls to EARLY for the sample problem case from almost 15 million to less than 0.5 million. An implementation of this revision was coded and tested by SNL but never implemented. SNL states that there were no savings in MACCS 1.5 run time on a VAX/VMS system with this revision.

There were two recommendations made relative to MACCS 1.5 based on the review of MACCS 1.5.0. It was strongly recommended that MACCS be placed under some kind of configuration control. This configuration management plan should contain a facility such as INCLUDE which would permit coding fragments to be identified by name and propagated throughout the coding automatically to specified program units. It is also strongly recommended that many of the common blocks in MACCS 1.5 be parameterized to facilitate program modifications using these commons. The following sections describe these two

recommendations and provide some examples from MACCS 1.5.0 for illustrative purposes.

The INCLUDE facility permits one to encapsulate a program fragment and to name it. The encapsulated

fragment can then be automatically propagated to specified program units. This would permit, for instance, a single copy of a common block to be kept. If this block is modified, the modified copy of the block would be automatically propagated to those program units which should contain it.

Table 6. Timing profile of the MACCS 1.5.0 sample problem calculation for the ten program units requiring the most time

<u>Routine Name</u>	<u>Execution Time (s)</u>	<u>Percent Time</u>	<u>Number of Times Called</u>	<u>Average Time per Call</u>
EFFGET	210.869	32.13	14,755,456	0.000
OUTPT1	204.210	31.12	488	0.418
DO1CDF	28.360	4.32	1,006,256	0.000
READ2	22.977	3.50	2	11.489
OXTPT9	19.612	2.99	244	0.080
DOSGET	16.129	2.46	1,102,720	0.000
CASGET	15.391	2.35	469,184	0.000
GETIMP	14.292	2.18	868,032	0.000
OXPT10	9.727	1.48	244	0.40
ECCGET	9.140	1.39	551,360	0.000

Table 7. Timing profile of the MACCS 1.5.0 sample problem calculation for the ten program units called the most times

<u>Routine Name</u>	<u>Execution Time (s)</u>	<u>Percent Time</u>	<u>Number of Times Called</u>	<u>Average Time per Call</u>
EFFGET	210.869	32.13	14,755,456	0.000
DOSGET	16.129	2.46	1,102,720	0.000
DO1CDF	28.360	4.32	1,006,256	0.000
GETIMPP	14.292	2.18	868,032	0.000
ECCGET	9.140	1.39	551,360	0.000
CASGET	15.391	2.35	469,184	0.000
EDOSIN	3.117	0.47	101,092	0.000
CLSHIN	1.859	0.28	73,330	0.000
POL2	1.435	0.22	73,330	0.000
AREA	0.924	0.14	72,184	0.000

This makes common block changes easier and much less error prone since all instances of the common block are changed automatically. For example, the

following program fragment occurs identically in eight different program units in MACCS 1.5.0:

```

PARAMETER (MAXREC = 1000, MAXLEN = 101)
CHARACTER * (MAXLEN) CARD, CLOC
COMMON /INPRC1/ CARD (MAXREC), CLOC
LOGICAL CRDFLG
COMMON /INPRC2/ IPNT (MAXREC), CRDFLG (MAXREC)
COMMON /INPRC3/ NBLANK, NCHANG, NCMMNT, NDPLCT, NREC, NRECT,
1          NTRMNT.

```

If this coding fragment were maintained under some kind of INCLUDE facility, then any change (to MAXREC, for example) would automatically be propagated to all eight program units. When maintained by hand as it currently is in MACCS 1.5., it is very easy to miss changing one or more program units.

While some of the common blocks in MACCS 1.5.0 are parameterized, as in the previously cited program fragment, most of them are not. It is recommended that common blocks be parameterized where possible and that this parameterization be propagated into the coding of the program units where possible. For example, MACCS 1.5.0 contains the following coding fragment:

```

INTEGER PARENT
REAL LAMBDA
COMMON /ISOGRP/ MAXGRP, PARENT (150), IGROUP (150),
$          HALFLIF (150), LAMBDA (150)

```

This program fragment could be parameterized as follows where the INCLUDE facility is done using COMDECK from UPDATE:

```

*COMDECK ISOGRP
  PARAMETER (LNA = 150)
  INTEGER PARENT
  REAL LAMBDA
  COMMON /ISOGRP/ MAXGRP, PARENT (LNA), IGROUP (LNA),
$          HALFLIF (LNA), LAMBDA (LNA)

```

This would permit this coding fragment to be distributed, by the name ISOGRP, to specified program units. If any of the program units used the number 150 in a way related to the array sizes, for a loop control for instance, that number could be replaced by the parameter LNA. In this way, whenever changes are required to the array size in ISOGRP, it is only necessary to change the parameter and this change will be automatically propagated to all specified program units by the the code configuration management program used.

SNL agreed that these two recommendations were

important. They did adopt a configuration control procedure based upon an UPDATE emulator on their VAX computer system. Changes made to MACCS 1.5 are well documented with all the specifics related to the existing problem, the proposed fix, and a test plan to ensure that the new code version works as desired. The changes were then sent out as an UPDATE correction set with the revision number and date included in the code and printed at the top of each output page. SNL has not yet parameterized the common blocks although they have stated that it should be done when the common blocks are restructured as discussed earlier.

2.2 Atmospheric Transport, Dispersion, and Deposition Routines

The following documents the findings of the QA and verification of the ATMOS program module of MACCS 1.5. The ATMOS program module calculates the atmospheric transport, dispersion and deposition of the released radioactive material. The QA and verification of the ATMOS program module involved comparing both the relevant documentation and FORTRAN code routines with cited references to assure proper model implementation. The following summarizes the documentation and code routines checked and the items identified as errors or inconsistencies.

The specific documentation used for the ATMOS QA and verification included Chapter 2 of the MACCS 1.4 Reference Manual and Chapter 1 of the MACCS 1.5 draft User's Guide. The Reference Manual provided the reference list and description for the atmospheric dispersion model used in MACCS 1.5. The User's Guide provided some additional model descriptions, the default data listings provided with MACCS 1.5, variable descriptions, allowed variable ranges and additional reference information.

The FORTRAN coding was reviewed in the order in which information is processed by MACCS 1.5 for the ATMOS program module. ATMODL, the ATMOS user input model description subroutine, ATPROB, the ATMOS user input problem description subroutine and ATMOUT, the main subroutine for the ATMOS module, were reviewed line-by-line as were all the externally referenced routines. The subroutines and functions comprising the ATMOS program module that were reviewed are listed in Table 8 with a short description of their purpose. The only external routines called by these program elements not reviewed were

CGET1, IGETN, IGET1, LGETN, LGET1, RGETN, RGET1 and ERRLOC, identified as part of the INPRE free-format input package. These routines are common to all of the MACCS 1.5 program modules and are discussed in Section 2.6.

Chapter 2 of the MACCS 1.4 Reference Manual was reviewed relative to the cited references. During this review, a number of inconsistencies were discovered. These were identified and discussed with the SNL MACCS code development staff for incorporation of corrections into the final documentation. The inconsistencies discovered were:

1. The equation on page 2-5 had a 2 in the denominator that does not belong based on Equation 3.4 on page 6 of Ref. 10.
2. The equal sign did not belong in the equation on page 2-6 between the Gaussian concentration for an infinite extent and the exponential terms for multiple reflections (Equation 5.8 on page 36 of Ref. 10).
3. The equation on page 2-7 represented $\chi(x,0,z;H)$ instead of $\chi(x,y,z;H)$. Based on Equation 5.9 on page 36 of Ref. 10, the denominator should have contained $\sqrt{2\pi}$ instead of 2π .
4. References in Chapter 2 with the prefix "US" were listed in the references with the prefix "NRC."

SNL provided the following corrections to the MACCS 1.4 Reference Manual to correct errors in model descriptions relative to the ATMOS program module in MACCS 1.5. These corrections were incorporated into later drafts of the MACCS 1.5 Model Description. The corrections identified by SNL were:

Table 8. Subroutines and functions reviewed during the ATMOS program module QA and verification

Subroutine or Function	Description
ATMODL	ATMOS input model description
INPGEO	Defines the geometric grid
INPWET	Loads the wet deposition data
INPDY	Loads the dry deposition data
INPDIS	Gets the dispersion parameter data
INPEXP	Defines plume expansion factor parameters
INPLRS	Defines plume rise model scaling factors
ATPROB	ATMOS input problem description
INPWAK	Defines dimensions for building wake effects
INPREL	Defines plume release description

Table 8. (continued)

Subroutine or Function	Description
PUTSTM	Stores and fetches source term data when more than one source term is being used
INPMET	Defines the weather sampling characteristics
INPM1	Defines the meteorological sampling option
WRDMET	Reads and packs weather data file
INPM2	Defines boundary weather conditions
INPM3	Defines accident start time
INPM4	Defines Latin hypercube sampling parameters
WBNMET	Groups meteorological data into bins
WNRZB	Determines windrose from binned meteorological data
INPM5	Defines 120 hours of user-supplied weather
INPOPT	Defines output options for ATMOS
ATMOUT	Models dispersion and deposition for each weather sequence
AREA	Calculates the area under a line segment
CAUGHT	Determines if plume trapping occurs
VELADJ	Adjusts wind speed to desired height
WASHOU	Calculates fraction remaining in plume after wet deposition
FSGY	Calculates σ_y at a given distance and stability class
FSGYIN	Calculates virtual source distance to a plume of given σ_y
FSGZ	Calculates σ_z at a given distance and stability class
FSGZIN	Calculates virtual source distance to a plume of given σ_z
DECAY	Calculates radioactive decay of release
PLMRIS	Calculates change in plume height over a given distance
SIGTEX	Provides text value of σ_z or "UNIFORM"

1. MACCS 1.5 does not test for $\sigma_z >$ the lid height as indicated on page 2-7. The code uses uniform mixing when that model produces a higher ground concentration than does the multiple reflections calculation. MACCS 1.5 may calculate a value of σ_z that exceeds the lid height.
2. The equation given on page 2-26 for material lost over the interval should read $\Delta Q = Q * [1 - \sum (B(i) * F_d(i) * F_w)]$.
3. The denominator for the average ground concentration C_g on page 2-27 should have a $\sqrt{2\pi}$ instead of a 2π term in it. No reference was provided for this equation in subsequent reviews of the references and the User's Guide. SNL did provide a more detailed derivation of this equation in subsequent drafts of the Model Description.
4. The termination of buoyant plume rise was not based on item 2 on page 2-31 ($2.15 \sigma_z =$ lid height).

The following items were identified as either coding errors in the MACCS 1.5 FORTRAN coding or inconsistencies between the coding and the MACCS 1.5 User's Guide. The list was compiled during the line-by-line verification of the ATMOS program module. The errors and inconsistencies discovered are presented in the order in which information is processed by MACCS 1.5. Each of these items was identified to and discussed with SNL MACCS code development staff. Unless noted otherwise, SNL made the correction as stated prior to the NUREG-1150 revisions. The items are:

1. The default dispersion parameter data given in the User's Guide on pages 17-18 were not consistent with those found in Ref. 11. The σ_y linear term for D-class stability should be 0.1474 instead of 0.1471. The σ_z exponential term for C-class stability should be 0.8543 instead of 0.8534.
2. The minimum given in the User's Guide for the allowable range of XPFAC1 (page 20) and XPFAC2 (page 21) should be 0.01 and not 0. to be consistent with the coding in subroutine INPEXP. The coding appears to be

correct in this instance and the User's Guide needed correction.

3. The word "dose" on page 22 under the discussion of the purpose of variable SCLCRW should have been "done."
4. The code common blocks have been expanded to allow up to 150 nuclides to be specified. Subroutine DECAY, however, aborted if the number of specified nuclides exceeded 60. SNL solved the problem by removing the IF-THEN test in DECAY that checked the number of nuclides.
5. The comment block in DECAY states that the subroutine is called by ATMOUT. It is also called by INPREL in MACCS 1.5.
6. INPOPT allows values of IDEBUG between 0 and 8 inclusive. The User's Guide listed the range as between 0 and 3 inclusive. In MACCS 1.5, values of IDEBUG 3 or greater produce the same debug output. SNL opted to leave the range alone and changed the User's Guide.
7. The comment block in CAUGHT stated that LIFMOD = .FALSE. will prevent plume entrainment. This variable was not used here or in ATMOUT, the calling routine. The comment statement was not removed from CAUGHT in the released version of MACCS 1.5.
8. No reference was given for computing the plume dimensions PLSIGY and PLSIGZ based on the building dimensions. These equations were found on pages 32 - 34 of Ref. 10. SNL expanded the write-up to describe the basis for PLSIGY and PLSIGZ in the Model Description instead of using the reference.
9. In the 170 loop of ATMOUT, the code was calculating the fraction of the plume that is in the interval during a given weather hour for wet deposition. When 120 hours of weather data have been used, the calculation uses boundary weather. The upper bound distance was recomputed using boundary weather wind speed and the time required for the remainder of the plume to leave the interval. The lower bound was not changed and remained at its previous value. The lower bound should be set to the previous value of the upper bound to prevent scavenging too much material from the plume over the interval. A potential fix was to move the 170 label up to the line calculating the lower bound of the interval. In discussions with SNL, it was found that the potential exists in the current coding logic for a divide by zero with this fix if the plume leaves the spatial interval at the same time boundary weather is initiated. SNL revised the methodology for handling boundary weather following the NUREG-1150 revisions and solved this problem at that time. For the NUREG-1150 revisions, an approach was devised to circumvent this problem. The approach used is discussed under item 10.
10. During discussions concerning ATMOUT with SNL, it was noted that the LIMSPA parameter that forces boundary weather beyond a specified interval was not being used in ATMOUT prior to the wet deposition calculation. This meant that the code was not forcing boundary weather in the last spatial interval, a feature planned for use in the NUREG-1150 revisions to force rain and, therefore, complete plume wet scavenging before the plume left the computational grid. Discussions with SNL culminated in the decision that the complexity of this portion of ATMOUT made an easy fix that would not adversely affect other coding virtually impossible. The most expeditious way around the problem was to make sure that the computational grid used for NUREG-1150 calculations was large enough to force boundary weather in the last grid element and to set LIMSPA equal to the element preceeding the last element. To ensure complete scavenging of the plume, a low wind velocity (0.5 m/s) would be specified for boundary weather. Since the plume will still be within the last element when we run out of weather hours and boundary weather is initiated, the divide by zero problem (item 9) will also be avoided. The boundary weather handling was revised in MACCS following the NUREG-1150 revisions to allow correct transition to boundary weather.
11. In ATMOUT, the stability, ISTAB(ISTOP), was compared to the previous weather hour's stability, ILAST, to determine if a call to FSGYIN (and FSGZIN if reflections are still being computed) was required to recompute a virtual source distance. The code did not,

however, reset the value of ILAST (initially set to 0) and the test always failed. This did not produce an erroneous result but, instead, forced the code to perform unnecessary computations. The fix was to add a line setting $ILAST = ISTAB(ISTOP)$ before the weather hour was incremented ($ISTOP = ISTOP + 1$). This inconsistency was corrected after the NUREG-1150 revisions.

12. An error was discovered in the the computation of reflection in the air and ground χ/Q exponential terms in ATMOUT. The 300 loop of ATMOUT was designed to compute up to 30 additional exponential terms, stopping when the additional terms add less than 1% to the current χ/Q exponential term. The test for convergence had the same statement label (350) for either outcome on the test and always exited the loop after the first pass. Reference 10 states on page 36 that 3 or 4 terms were sufficient to include the important reflections. SNL fixed the problem prior to the NUREG-1150 revisions by removing the convergence test from ATMOUT and evaluating only the first 5 exponential terms.
13. No reference was found for the calculation of effective height (EFHGHT) following the loop on reflection terms (loop 300). SNL remedied this by providing a detailed discussion of the plume rise model variables in the Model Description.

During the transition from MACCS 1.5.0 to the released version of MACCS 1.5, changes were made to ATMOS to correct problems found by SNL. The first error was in ATMOUT where, on 32-bit computers, the ground concentration was incorrectly reported as zero due to roundoff errors. The variables involved were changed to double precision and the problem was fixed prior to the NUREG-1150 revisions. ATMOUT was also modified to provide correct timing of the plume when the transition to boundary weather occurred. SNL had noted the timing errors when investigating the previously documented errors in this part of ATMOUT. This error was fixed after the NUREG-1150 revisions. Finally, SNL changed the upper bound on PDELAY in INPREL from 604,800 s to 345,600 s to prevent a code error when the time between the release of the first plume and the reference point of the last plume exceeds 120 hours. The correction was made after the NUREG-1150 revisions.

SNL and the INEL reviewer had numerous discussions concerning the overall complexity of the ATMOUT subroutine. ATMOUT uses numerous time frames and variables to track various parts of the plume relative to the beginning, middle and end of the current spatial interval. Many of the important variables used in ATMOUT are not defined in the comment block at the beginning of the routine. Some variables are defined to be a specific parameter but are later used for other purposes and no longer represent their original definition. For instance, ISTOP is defined to be the index of the last hour of weather in an interval but is instead used as a weather hour loop counter in ATMOUT. ATMOUT calculates plume decay parameters, plume growth, plume rise parameters, wet and dry deposition parameters, air and ground concentrations, plume depletion and debug printout output parameters for the ATMOS program module. It is difficult to determine the various model and process interdependencies because of the large number of interrelated variables.

A proposed method of avoiding this complexity would be to break ATMOUT up into discrete subroutines and functions to handle each of these calculational steps separately. SNL proposed a main subroutine looping over spatial intervals that would call separate routines to compute plume timing variables, plume decay, plume growth, plume rise, dry deposition, wet deposition, air and ground concentrations, plume depletion due to scavenging, and the parameters needed for the debug printout. Restructuring ATMOUT would not only facilitate error correction but would also facilitate future model revisions. ATMOUT was not rewritten for the released version of MACCS 1.5 with the exception of fixing the aforementioned errors and inconsistencies.

The coding that comprises the ATMOS program module appears to be correct relative to the cited references in the released version of MACCS 1.5. The items identified relative to the Reference Manual and User's Guide were incorporated into the final documentation as stated. The only outstanding item is the restructuring of the ATMOUT subroutine. As stated above, restructuring of this routine will aid in future code modifications.

2.3 Early Consequence Routines

The following documents the findings of the QA and verification of the EARLY program module of MACCS 1.5. It is the function of EARLY to calculate the health effects of a radioactive release due to

exposures that occur in the first seven days after the accident. EARLY relies on information from the atmospheric models (ATMOS) in the form of air and ground concentrations of a set of radionuclides as a function of position and time. The QA and verification of the EARLY program module involved comparing both the relevant documentation and FORTRAN code routines with cited references to ensure proper model implementation. The User's Manual used during this review was not checked against the code for consistency relative to the variable specifications. This task was performed during the final review of the draft User's Manual and is documented in Section 3. The following summarizes code routines checked and the items identified as errors or inconsistencies.

The executive routine of the EARLY module is called EAROUT. This routine first calls the routines EGEOM and EPCALC for the first emergency response strategy (the loop on emergency response strategies is outside of EAROUT). These routines calculate the Gaussian plume geometry factors, the cloudshine correction factors and the early dose conversion factors. EAROUT then calls RELZON which calculates doses outside of the emergency response zone, ESTAT which calculates doses inside of the emergency response zone, and EMOVE which calculates doses during evacuation from the emergency response zone. The risk of early fatalities is then calculated in FATRIS, the risk of early injuries in INJRIS and the risk of cancers from early exposures in CANRIS. Finally, EAROUT calls STOEAR which calculates the set of results prescribed by the user and writes them to a file.

A review was performed on the nine major routines of the EARLY module summarized above and the subsequent daughter routines. All of the EARLY module routines that were reviewed are listed in Table 9 with a short description of their function. Overall, the routines were found to be correctly written and with a good coding style that was easy to follow and understand. The following summarizes the findings and recommendations resulting from this review. As before, the status of the recommendations relative to both the NUREG-1150 revisions and the released versions of MACCS 1.5 is also provided.

The subroutine EGEOM calculates the Gaussian distribution that is used to determine doses away from the release plume centerline. A cross wind dispersion factor (σ_y , calculated from user input) characterizes the plume dispersion perpendicular to the direction of the wind. The dispersion is limited to $2.15 \cdot \sigma_y$. The perpendicular Gaussian distribution of plume concentration is mapped onto a polar coordinate system which distorts the Gaussian distribution somewhat. If σ_y is greater than $0.31 \cdot r$ (r = radial distance from release point), the arc length that the perpendicular distribution is mapped onto becomes less than 90% of the perpendicular distance. However, σ_y is usually not greater than $0.25 \cdot r$ even in weather favorable to dispersion, so the distortion is not expected to have any significant impact unless an inordinately large value of σ_y is used. The effect on the NUREG-1150 revisions should be minimal and would, in all cases, overpredict the plume concentration and resulting dose. This was not deemed a problem but only an observation. SNL states that the approach chosen was selected with full knowledge of its resultant small numerical error.

Table 9. Subroutines and functions reviewed during the EARLY program module QA and verification.

Subroutine or Function	Description
EARINP	EARLY input model description
INMISC	EARLY miscellaneous input
INORGA	Defines the organ list used in EARLY
INEVAC	Loads the wet deposition data
INOPU	Defines the site population distribution
CMPTBL	Checks number of items on site file
MATCH	Checks key separator on site file
EVRADI	Reads the evacuation data
INPEMR	Defines the emergency response zone
INDFAC	Reads protection and exposure factors
INEFAT	Defines the EARLY fatality risk model
INEINJ	Defines the EARLY injury model
INACAN	Defines the acute cancer model
INOUT1	Defines the options for result number 1

Table 9. (continued)

Subroutine or Function	Description
INOUT2	Defines the options for result number 2
INOUT3	Defines the options for result number 3
INOUT4	Defines the options for result number 4
INOUT5	Defines the options for result number 5
INOUT6	Defines the options for result number 6
INOUT7	Defines the options for result number 7
INOUT8	Defines the options for result number 8
REDSTG	Loads emergency response data for multiple strategies
PUTSTG	Stores all evacuation strategies in list storage
GETSTG	Reads an evacuation strategy from list storage
OUTCON	Writes header record on EARLY and CHRONC output files
HEDEAR	Prepares the results file for EARLY
RESNM1	Returns name of requested type 1 effect
DISRAN	Returns a distance range for results
DIST1	Returns a distance for results
COMPRS	Compresses blanks in result names
RESNM2	Returns name of requested type 2 effect
RESNM3	Returns name of requested type 3 effect
RESNM4	Returns name of requested type 4 effect
RESNM5	Returns name of requested type 5 effect
RESNM6	Returns name of requested type 6 effect
RESNM7	Returns name of requested type 7 effect
RESNM8	Returns name of requested type 8 effect
EAROUT	Calculates EARLY results for a single weather trial and emergency response strategy
EGEOM	Calculates plume geometric factors for EARLY
CLSHIN	Returns the cloudshine correction factor
POL2	Performs bilinear interpolation from a table
EPCALC	Precalculates EARLY dose conversion factors
RELZON	Calculates doses outside of the emergency response zone
ESTAT	Calculates doses to stationary individuals inside the emergency response zone
CENACU	Accumulates centerline doses in arrays
CENZER	Zeros out the centerline dose arrays
EDOSIN	Calculates centerline doses
INCDOS	Accumulates doses to stationary individuals over the fine grid
EMOVE	Accumulates doses to moving individuals over the fine grid
ZERREM	Zeros out accumulated dose for elements which require hot spot or normal relocation
INCREM	Recalculates dose for relocated individuals
FATRIS	Calculates EARLY risk of fatality
INJRI	Calculates EARLY risk of injury
CANRIS	Calculates EARLY risk of cancer
STOEAR	Controls calculation of EARLY results
OUTPT1	Calculates result number 1
EFFGET	Helper routine for OUTPT1
OUTPT2	Calculates result number 2
OUTPT3	Calculates result number 3
OUTPT4	Calculates result number 4
OUTPT5	Calculates result number 5
OUTPT6	Calculates result number 6
OUTPT7	Calculates result number 7
OUTPT8	Calculates result number 8

The subroutine CLSHIN calculated a cloudshine correction factor using a table of dose correction factors taken from p. 18 of the January 1988 draft copy of Ostmeyer's unpublished report on exposure pathways.^a Based on the reference, however, one of the values was incorrect in the code: 0.066 instead of 0.0066 for effective cloud sizes of 0 and 3 meters at an effective distance of 3. The table contained in MACCS was taken from and reflects the same error found in Ref. 12, Table VI 8-1, p. 8-4. This is not considered a serious error since plumes generally start with an effective size of 30 meters and grow from there. For the sake of accurate coding, however, the error was fixed prior to the NUREG-1150 revisions.

The subroutine EPCALC calculates the parts of the dose conversion factors that require looping over the number of radioactive nuclides in the source term. This is done here for all five exposure pathways to reduce the number of times this looping is required. These factors are later used in subroutine EDOSIN to calculate a dose for a particular time period. The groundshine exposure pathway is modeled in two steps: groundshine during plume passage and groundshine after plume passage. The groundshine after plume passage decays with time according to the decay constants of the various nuclides. However, in order to avoid looping over all the nuclides for each time period a dose is required, a single effective decay constant (λ_{ef}) for the plume inventory of nuclides is calculated. This is done by calculating the dose for 8 and 168 hrs (7 days), taking the ratio of these two doses and then interpolating λ_{ef} from a table of solutions to the dose ratio for a single decay constant. The solution approaches infinity as the ratio approaches 1 and becomes non-physical below a ratio of 1/21 (8/168) as λ_{ef} becomes negative.

To test the accuracy of a single effective decay constant, a two component source term was devised with one half life set at 4 hrs and the other half life set at 300 hrs. The effective decay constant was calculated as in MACCS 1.5 by taking the ratio of doses from the sum of two components at 8 and 168 hrs and interpolating the solution. The approximation is exact at 8 and 168 hrs, as expected, and diverges somewhat at other times. When the long half life species dominates (ten to one) the approximation becomes almost exact at all times.

a. R. M. Ostmeyer et al., *Exposure Pathways Models for Accidental Radionuclide Releases*, NUREG/CR-4184, Sandia National Laboratories, to be published.

The models for all other exposure pathways (including groundshine during plume passage) are simple enough that they can be coded exactly without looping over nuclides outside of EPCALC. No quantitative assessment of the single effective decay constant was made relative to the error that would occur for the NUREG-1150 revisions. Qualitatively, it was observed that in all cases the simplification will overstate the ground concentration and therefore the dose. SNL states that the use of a single decay constant was an intentional choice and not an error.

The subroutine RELZON calculates the dose received by individuals outside of the emergency response zone. It first calculates the dose received for the entire "early" (typically 7 days) phase of the accident and then recalculates the dose assuming relocation for those area elements that exceed a relocation criteria. When the dose is recalculated, subroutine RELZON loops over plume segments (typically 2 for NUREG-1150 type calculations) calculating the centerline dose from each segment and then applying the Gaussian distribution on the polar grid. However, in this last step MACCS 1.5 set the dose to the current plume value rather than summing over all plume segments. This was an error that was deemed critical to the NUREG-1150 revisions. SNL determined that the error actually occurred in subroutine INCREM. SNL rewrote INCREM to fix this error and to add a glossary of variables to the subroutine prior to the NUREG-1150 revisions. The revised routine was reviewed and determined to be correct.

The subroutine ESTAT calculates the dose received inside the emergency response zone, and contained a similar error as that found in RELZON. ESTAT branches depending on which response zone is being considered. For either of the two sheltering zones ESTAT calculates the dose for a pre-sheltering period and a post-sheltering period. However, the centerline doses are incremented after the post-sheltering, but not after the pre-sheltering period. This means that there was no dose (or risk) for normal activity that preceded sheltering. For the three evacuation zones the doses were calculated in one step and the centerline doses were incremented correctly. SNL fixed this problem by writing a new routine CENACU to accumulate centerline doses with an entry point CENZER to zero out the centerline dose arrays. SNL also removed in-line coding in EMOVE and INCREM that handled the accumulation of centerline doses with calls to the new routine. The new routine and changes to existing routines were reviewed and found to be correct.

The subroutine EMOVE calculates the doses received during evacuation from the emergency re-

sponse zone. However, individuals in the two sheltering zones are not considered by this model, and therefore do not accumulate any dose during evacuation as do the individuals in the three evacuation zones. This is how the code is designed now, but it could be changed by generalizing EMOVE to consider all of the emergency response zones. A realistic emergency response would be to first shelter all zones and then evacuate critical regions. The released version of MACCS 1.5 still treats shelterees and evacuees separately and does not allow for shelterees to accumulate dose when relocated.

The three risk routines (FATRIS, INJRIS, and CANRIS) were straightforward and no errors were found. The risk of injuries is calculated independent of the risk of death. So, death is a subset of injury and the load on hospitals would be determined from the injury risk rather than the sum of injury and death risk. However, the risk of latent cancer is calculated dependent on the risk of death, so the risks of acute cancer and early fatalities are not subsets of each other.

Any routine within the EARLY module (not including input routines) not specifically mentioned was deemed straightforward and no errors were found. The review has determined that the EARLY program module routines were correctly written and that the programming style made them easy to follow and understand.

The one aspect of EARLY not reviewed was the method by which the dose conversion data, used by both EARLY and CHRONC, are compiled for use by MACCS 1.5. As stated in the User's Guide, the data is obtained from Oak Ridge and preprocessed at SNL. The preprocessor program aggregates the data into a MACCS readable file and also applies "effective acute" dose conversion factors. This preprocessor was

received by INEL but due to a lack of internal or external documentation could not be reviewed. This preprocessor should be reviewed to ensure accurate preparation of the dose conversion data. It should also be provided with the MACCS 1.5 code along with complete documentation to allow users the ability to create their own data sets for organs and isotopes not included in the default dataset transmitted with the code. SNL has indicated that the preprocessor will be included with appropriate documentation with MACCS 1.5.

There were errors uncovered by SNL in EARLY during the transition from MACCS 1.5.0 to the released version of MACCS 1.5. The first involved mapping of the doses onto the computational grid when the wind-shift option was used. SNL found two errors, one in INCDOS where the doses were mapped onto the grid incorrectly when the rotation reached direction 1 or 16. A second error was discovered in INCREM where mapping of doses for relocation was performed correctly only to the right of the plume centerline. SNL fixed both of these errors prior to the NUREG-1150 revisions. SNL discovered that an incorrect method was being used in INCREM for zeroing out previously calculated doses and accumulating new doses for shorter exposure periods. SNL found that this only gave correct answers if all plume segments went in the same direction and had the same release duration. SNL separated the two functions by creating subroutine ZERREM and fixing the dose accumulation in INCDOS. These corrections were made prior to the NUREG-1150 revisions. SNL discovered an error in ESTAT where incorrect flag values for RETCOD were being calculated in the outer shelter zone. SNL fixed this problem after the NUREG-1150 revisions. SNL added a feature to EARLY that allowed the user to specify "acute" or "lifetime" dose for population exceeding a dose in INOUT3 after the NUREG-1150 revisions.

2.4 Chronic Consequence Routines

The following documents the findings of the QA and verification of the CHRONC program module of MACCS 1.5. The CHRONC module calculates the long-term economic and health effects resulting from a radioactive release. The time frame of the CHRONC module's calculations are from the end of the EARLY phase to infinity. The line-by-line verification included all of the input, main processing, and output routines that make up the CHRONC program module. Because the documentation of CHRONC was not complete, the review required a walkthrough of the MACCS 1.5 CHRONC program module with the SNL code developers. The January 1988 draft copy of Ostmeyer's unpublished report on exposure pathways and the July 22, 1988 draft copy of the MACCS 1.5 User's Guide were also used. The following describes the results of this review as well as subsequent alterations made by SNL to the CHRONC module in MACCS 1.5.0.

The reviewed subroutines and functions that comprise the CHRONC module are listed in Table 10. The items reported to SNL as potential coding errors or discrepancies between the code and documentation follow. Resolution of each potential problem by SNL is also provided.

1. Subroutine STPATH, which reads and initializes the CHRONC ingestion pathway data, was called from both CHRINP and INPCHR. One of the calls is redundant, since only one call to this subroutine is required. SNL removed the redundant call in CHRINP prior to the NUREG-1150 revisions.
2. DO loop 370 in subroutine RDISTB and DO loops 280 and 320 in subroutine EXCINP all perform tests on the food pathways isotopes to check for input errors, but each one uses the water pathways isotopes list to obtain the food isotopes names. Since the food pathways isotope list is always a subset of the water pathways isotope list, this is not an error though it makes the coding hard to follow.
3. Format statement 180 in subroutine SDFINP, which reads the site data file, was not consistent with the format of the site data file watershed data shown in Appendix C of the draft MACCS 1.5 User's Guide. The error was corrected in the MACCS 1.5 User's Guide.
4. Subroutine EXCINP, which reads the dose factors conversion data file, has two places where the 2 line intermediate header at the beginning of the dosimetry data as specified by the User's Guide appears to be skipped. In fact, only one line is skipped at each read so that the code and documentation are consistent and there was no error.
5. In subroutine GNDRES, the calculation of the level of radioactivity of daughter nuclides from the weathering equation shows the decay constant of the daughter nuclide as a multiplicative factor. The number of daughter nuclei, N , as a function of time, t , is given by

$$N = [\lambda_p N_o / (\lambda_d - \lambda_p)] [\exp(-\lambda_p t) - \exp(-\lambda_d t)] \quad (1)$$

where λ_p is the decay constant of the parent nuclide, λ_d is the decay constant of the daughter nuclide and N_o is the initial number of daughter nuclei. Since the level of radioactivity is proportional to the number of nuclei, it appears that the calculation should instead show the decay constant of the parent nuclide as a multiplication factor. The same would be true for the calculation of the level of daughter nuclide radioactivity in subroutine DECAY. SNL pointed out that the equation in MACCS 1.5 is calculating activity, not the number of nuclei. The activity equation is given by:

$$\alpha = \lambda N \quad (2)$$

where λ is the decay constant and N is the number of nuclei. Hence, it is proper to show the daughter nuclide decay constant as a multiplicative factor in the daughter nuclide radioactivity calculations. Though the parent nuclide decay constant does not explicitly appear in the MACCS 1.5 FORTRAN statements, it is implicitly present in the other multiplicative terms. Thus, the calculations are correct. SNL expanded the discussion in the Model Description to describe the derivation of the equation found in MACCS 1.5.
6. Subroutine WTRTRF used the equation

$$W = [\lambda_a \lambda + \lambda_b(1 - \lambda_a)] / (\lambda + \lambda_b) \quad (3)$$

for the total washoff fraction of a deposited radionuclide. According to Ref. 13, page 751, equation 3.3, the correct equation should be:

$$W = (\lambda_a \lambda + \lambda_b) / (\lambda + \lambda_b) \quad (4)$$

This problem had been reported to SNL by some MACCS users in Spain and SNL had corrected it in Version 1.5.0. The version of 1.5 reviewed during the CHRONC review predated this correction. With this one exception, however, the CHRONC program module in the version reviewed and in Version 1.5.0 were identical.

The review uncovered a number of errors regarding input range and data format discrepancies between the code and the User's Guide. These discrepancies, listed in the Table 11, were determined to be errors in the draft User's Guide and not errors in MACCS 1.5. SNL

states that they have corrected these errors in the final version of the MACCS 1.5 User's Guide.

The few errors uncovered during the review would tend to indicate accurate programming of the referenced models. One of the subroutines, STPATH, was identified as a very complex routine in the coding analysis discussed in Section 2.1 and the CHRONC reviewer concurred with this assessment. However, since STPATH simply reads input and writes out data into arrays used by CHRONC, its function was easily checked and it was unwarranted to conclude that it needed to be rewritten.

SNL made other changes to CHRONC between MACCS 1.5.0 and the released version of MACCS 1.5. These added features and corrected errors discovered by SNL. Subroutines STPATH and RDISTB were modified to validate the ingestion pathways data prior to the NUREG-1150 revisions. The maximum number of allowable economic regions, NECRGN, was changed from 59 to 99 and the user input was validated to ensure it was valid in SDFINP following the NUREG-1150

Table 10. Subroutines and functions reviewed during the CHRONC program module QA and verification.

Subroutine or Function	Description
CHRIINP	CHRONC input model description
OPNERL	Copies modeling data from common blocks used by EARLY into those used by CHRONC
INPCHR	Control routine for reading CHRONC user input
INCHRN	Defines user input for CHRONC
STPATH	Reads and validates ingestion pathways data
RDISTB	Reads a table of ingestion pathway nuclide data
IXOT9	Defines the options for result type 9
IXOT10	Defines the options for result type 10
IXOT11	Defines the options for result type 11
IXOT12	Defines the options for result type 12
STTIMS	Sets time for the chronic and economic models
SDFINP	Reads and checks the site data file
CXPTBL	Checks number of items on site file
MXTCH	Checks key separator on site file
CKINDEX	Checks indices of integer array INTARR
EXCINP	Reads the CHRONC dose conversion factors
STGRDA	Defines site region characteristics when site data file is not used
CHRNDF	Computes all CHRONC time-independent dose factors
BLDTBL	Builds isotope daughter table
GNDRES	Computes groundshine or resuspension dose conversion factors
TRFRCT	Computes direct transfer fractions and long-term transfer factors
WTRTRF	Computes liquid pathway transfer factors
COPCHR	Sets up CHRONC to produce results requested by EARLY and CHRONC
HEDCHR	Prepares result file for CHRONC
RXSNM9	Returns name of requested type 9 result
RXNM10	Returns name of requested type 10 result
RXNM11	Returns name of requested type 11 result
RXNM12	Returns name of requested type 12 result

Table 10. (continued)

Subroutine or Function	Description
CHROUT	Calculates CHRONC results for a single weather trial
SGCPLN	Builds the straight line ground concentrations
WGCPLN	Builds the wind shift ground concentrations
CRNRSK	Determines the chronic risks for a single weather trial
DIRDEP	Computes pasture and crop transfer factors for directly deposited radionuclides
INITLZ	Initialize action and cost arrays
EMRGPH	Emergency phase economic risk model
INTRPH	Intermediate phase emergency response model
LNGTPH	Long-term phase dose and economic risk model
LTPROJ	Calculates long term actions required to meet criteria
LTMACT	Calculates long term actions required to meet habitability criteria
CSTEFF	Computes costs of projected actions and determine if their cost-effective
CSTDCN	Computes decontamination costs
LTACUM	Accumulates doses and costs resulting from long term actions
LOKSEE	Prints a summary of doses and costs for a spatial interval
STOCHR	Controls calculation of chronic/economic results
OXTPT1	Calculates result number 1
CASGET	Helper routine for OXTPT1
OXTPT4	Calculates result number 4
OXTPT5	Calculates result number 5
OXTPT6	Calculates result number 6
OXTPT7	Calculates result number 7
OXTPT8	Calculates result number 8
OXTPT9	Calculates result number 9
DOSGET	Helper routine for OXTPT9
OXPT10	Calculates result number 10
ECCGET	Helper routine for OXPT10
OXPT11	Calculates result number 11
OXPT12	Calculates result number 12
GETIMP	Helper routine for OXPT11 and OXPT12

Table 11. Errors detected in the draft MACCS 1.5 User's Guide during the CHRONC program module review.

Page Number	Description of error
101	Allowed Range for NUMWPI should be 1 < value < 10
101	Allowed Range for NAMWPI should be 3 < length < 8
103	Allowed Range for NAMIPI should be 3 < length < 8
115	Allowed Range for CTHALF should be 1. < value < 1.E14
118	Allowed Range for NAMIPI should be 3 < length < 8
118	Allowed Range for PSCMLK should be 1.E-20 < value < 1.E20
119	Allowed Range for PSCOTH should be 1.E-20 < value < 1.E20
119	Allowed Range for NAMIPI should be 3 < length < 8
120	Allowed Range for GCMAXR should be 1.E-35 < value < 1.E35
120	Allowed Range for QROOT should be 0. < value < 1.E35
123	Allowed Range for ORGNAM should be 2 < length < 10
123	Purpose for IX1DS9 references NUM9, should reference NXUM9
123	Purpose for IX2DS9 references NUM9, should reference NXUM9
147	Agricultural Data Block shows variable CROP in columns 15-25, should be columns 6-25
147	Watershed Data Block shows variable WSHIFR in columns 15-25, should be columns 16-25

Table 11. (continued)

Page Number	Description of error
147	Watershed Data Block shows variable WSHART in columns 25–34, should be columns 26–35
147	Watershed Data Block shows variable WTRINF in columns 35–74, should be columns 36–55
148	Regional Economic Data Block shows variable DPF in columns 25–30, should be columns 26–30

revisions. Variable ICHECK in SDFINP was increased from 6 to 10 to correctly implement 10 crop categories after the NUREG–1150 revisions. SNL added the capability to couple or decouple growing-season and long-term mitigative actions on food. This involved adding a new variable to the CHRONC input and modifying STPATH and LTPROJ. This was done after the NUREG–1150 revisions. SNL fixed an inconsistent application of radiological decay factors in TRFRCT for the long-term uptake pathway after the NUREG–1150 revisions. In subroutine EXCINP, SNL changed the array size of FILNUC, INDNUC, and IFDNUC from 100 to 300 to match the sister arrays in EDCINP after the NUREG–1150 revisions. Subroutine INPGRC was removed from the CHRONC module to prevent double-counting of intermediate phase radioactive decay after the NUREG–1150 revisions. SNL corrected an error that unintentionally tied the number of decontamination levels to the number of temporary interdiction periods in subroutine LTMACT. The correction involved fixing the number of interdiction periods at 3 as documented. This was done after the NUREG–1150 revisions. SNL discovered an error in ECCGET where the economic costs for EARLY are calculated. The variable TRMEVA appeared twice in the equation effectively double-counting the days-away-from-home cost. SNL fixed this error after the NUREG–1150 revisions. The allowable number of types of water bodies was increased from 2 to 4 in the code and the User's Guides after the NUREG–1150 revisions. SNL also cleaned up the internal documentation in the CHRONC module and rewrote STPATH and LTACUM to revise and clarify the logic after the NUREG–1150 revisions.

2.5 Meteorological Data Routines

The meteorological data routines in MACCS 1.5 process user supplied meteorological data and perform all requested weather binning and sampling. These routines were reviewed separately from the other modules

because the complexity of the program modules and the lack of adequate documentation made some program units very difficult to follow. A coding walkthrough with the SNL code developer responsible for this part of the MACCS 1.5 program was accomplished that clarified the programming logic to a point that allowed a detailed review. The following discusses the results of that review.

The majority of the meteorological data routines in MACCS 1.5 are devoted to sampling and binning the meteorological data. WRDMET is the subroutine that reads a years worth of hourly weather data from the meteorological data file. This routine is called from INPM1, where the variable METCOD is read. METCOD controls the MACCS simulation in the following way:

- METCOD=1 – User specified start day and hour – DAYHOU
- METCOD=2 – Weather category bin sampling – BINSAM
- METCOD=3 – User supplied weather data (120 hours) – USRSUP
- METCOD=4 – Constant weather conditions – CONMET
- METCOD=5 – Stratified random sampling – RANSAM.

The meteorological data file is read whenever METCOD is set to 1, 2, or 5. Subroutine WBNMET determines the bin category for each hour of weather data in the year and prints output summary tables. This routine is called from INPM4, where the data is loaded for weather sampling. WBNMET calls subroutine WNDRZB which computes the windrose from the meteorological data in the bins. For METCOD=2, subroutine BINSAM is used to perform the weather category bin sampling and is called from the main program, MACCS. BINSAM calls subroutine WRANBN to initialize the weather bin sampling strategy.

The balance of the meteorological data routines either read 120 hours of user supplied weather data, read 120 hours from the weather data file using a user supplied start day and hour, or read and set constant weather conditions. These routines were reviewed and found to be fairly straightforward with no errors uncovered. The following describes the findings of the review of the binning and sampling routines and associated external files. One important question was to determine the applicability of the random number generator, RANDOM, to the calculations involved in the weather category bin sampling. In the subsequent comments, this is addressed along with several other points from the detailed study of these routines. Comments are provided for each subroutines or function in which discrepancies, errors or potential programming improvements were noted. Resolution of these items by SNL is provided with each comment.

WRDMET: reads the meteorological data file, checking validity of data and printing summary.

Recommend that the array IDTA(24, 365) be used in common block /METDTA/ instead of MONTHS (12, 24, 31) and that the EQUIVALENCE of these two arrays be removed in this routine and in routines WBNMET and WGTMETT. MACCS 1.5 still contains these two arrays and the EQUIVALENCE.

Discussion with SNL indicated that the shift of the rain data by a factor of 10 at line 81 was an artifact of earlier code when the right digit was a flag to indicate rain or no rain. Although it still remains in MACCS 1.5, it should be removed to avoid confusing the logic in this subroutine.

The conversion of rain from hundredths of inches to tenths of millimeters and subsequent test to restrict MRN to be less than or equal to 999 places an implicit upper limit on the rain rate to approximately 3.93 in./hr. The User's Guide states that the limit is 9.99 in. of accumulated rainfall in an hour. The code or documentation should be changed to make the limits consistent.

The call to ERRFIL and the following WRITE statement indicated that the meteorological data file was UNIT 27, when, in fact, WRDMET was called from INP1 with a value of 28 for INMET, the meteorological data file unit number. This was corrected by SNL in the released version of MACCS 1.5.

WBNMET: this routine serves two purposes: first to categorize the yearly weather data by hour and second to print the bin summary report.

The report generation could logically be separated from the actual categorization process to improve the modularity of the code and hence the readability. SNL agrees that this routine could use rewriting to clarify the logic but has not done so. This applies to the following comments about WBNMET also.

The initialization of IWGHT would be more appropriately done in INPM4.

The value of ILST set in a DATA statement was meaningless, if not misleading, since it was set early in the routine to be NBIN+1.

The constant used to convert from tenths of meters per second to kilometers per hour was incorrect: 2.7764 is not the value of $(10 \times 1000 / 3600)$, which is the conversion factor. This occurred in both this routine and in INPM4. SNL changed the conversion factor to the exact value in both WBNMET and INPM4 prior to the NUREG-1150 revisions.

Within the loop on 80, the value of IRTBN was incorrectly set. The code was using variable NRNINT and should use NRINTN. The variable NRINTN refers to rain intensity and NRNINT refers to the distance at which rain occurs. SNL had discovered this error preparing for the coding walkthrough. This problem was compounded by the similarity in spelling of the two variables and further went undetected, because in the sample runs, the values of both variables were the same, 4. A similar error was uncovered during subsequent review of WBNMET. In the two places where MET is calculated, variable NRINTN was being used whereas NRNINT should be used. SNL fixed all of these errors just prior to the NUREG-1150 revisions.

At line 343, the bin categorization code is packed into the word containing the rain data.

A restructuring of this routine would further aid in the understanding of how the bin categorization is performed. It was recommended that alternative code using DO loops and IF constructs be used to simplify the logic in this routine.

WNDRZB: computes the windrose from the meteorological data in the bins.

A comment could be added to indicate that the ROSEBI array is constructed using the transpose of

portions of the IRDB and IRDBIN arrays. As with WBNMET, SNL has decided to rewrite this routine at some future date to clarify the logic and thus has opted not to clarify the current coding. This applies to the following comments as well.

The initialization of ROSEBI in the DO 2 loop should also include the last column of the array, J=NBIN+1 instead of J=NBIN.

The interface between WBNMET and WNDRZB could be simplified. Both ISMW and ISMRY are defined in WBNMET, with ISMRY=ISMW+1. Only the product of NRNINT and NRINTN is used in WNDRZB.

BINSAM: performs the weather category bin sampling.

The sampling procedure is initialized by calling WRANBN. An interval is set in SPACE as:

$$N / K$$

where N = # samples in category, and K = # samples to be chosen in the category for each bin (I=1 to NBIN). The random number generator is called and an integer IRAND is selected for each bin. The FORTRAN statement in WRANBN for the initial value is

$$IRAND(I) = XYZ * SPACE(I) + 0.9999$$

allowing the integer value to be between 0 and the greatest integer in the number N/K. What was intended

$$IRAND(IMET) = SPACE(IMET) * (XYZ + IBIT(IMET)) + .9999$$

gives an integer between $INT(IBIT(IMET) * SPACE(IMET) + .9999)$ and $INT((1+IBIT(IMET)) * SPACE(IMET) + .9999)$, or with $J=IBIT(IMET)$ and $(N/K)=SPACE(IMET)$, the value is between $INT[J * (N/K) + .9999]$ and $INT[(J+1) * (N/K) + .9999]$. Whereas the algorithm described in Ref. 14 would calculate the next value as an integer between $INT[J * (N/K)] + 1$ and $INT[(J+1) * (N/K)]$.

In the example in Ref. 14, a category contains 10 weather sequences from which four are to be sampled. The four sets contain 2, 3, 2, and 3 sequences respectively. The current implementation leads to such erroneous samples as: 3 3 7 10, or 0 3 7 9, or 3 5 8 10. SNL

was for IRAND to be between 1 and the greatest integer in N/K. This can be accomplished by replacing the above line of code with:

$$IRAND(I) = 1 + XYZ * (INT(SPACE(I) - 1)).$$

SNL changed the statement to:

$$IRAND(I) = 1 + (.999999 * XYZ) * INT(SPACE(I)).$$

This change was required as the term .999999 * XYZ must be a uniformly distributed random number, X, within the range $0 \leq X < 1$.

The simulation is started at day 152, hour 1. The Latin hypercube selection criterion is tested for each successive hour until all simulations have been performed. The test is if the category for the current hour, retrieved in IMET from the array MRAIN containing the packed format of the weather data, is

7. To be sampled at all (IWGHT > 0)
8. The current data point is the one in the category pre-selected by the random number generator (IBIN = IRAND)
9. Not all samples from the current category have been selected yet (IBIT < IWGHT).

When successful, the random number generator is called again and the next selection point is set in IRAND. The FORTRAN statement

pointed out that BINSAM is coded to prevent selecting the same sequence twice or selecting the zeroth element. The coding selects the next available sequence so that the actual selection in the first set would be 3 4 7 10. This implicitly shifts the third sequence from set 2 to set 1.

This problem was considered by the reviewer to be serious, especially for categories with relatively small populations and a moderate number of sequences to be chosen. Uncorrected, it allows values not belonging to a set to be chosen.

The following code will correct the problem:

```

J = IBIT(IMET)
ILEFT = INT( J * SPACE(IMET)) + 1
IRGHT = INT ( (J+1) * SPACE(IMET))
IRAND(IMET) = ILEFT + XYZ * (IRGHT - ILEFT + 1).

```

SNL adopted this change with one minor revision. The variable XYZ is multiplied by .999999 so that the random number, X, represented by this product is in the range $0 < X < 1$.

RANSAM: performs the stratified random sampling (METCOD=5). Although not part of the weather binning and sampling algorithm, it also uses the random number generator, RANDOM, and is included for completeness of the discussion of the random sampling techniques.

RANSAM uses the value of NSMPLS which is set by routine INPM4 in the common block /M4/ to determine how many trials to sample per day. A test on NSMPLS in RANSAM restricts its value between 1 and 4. However, in INPM4, the input value is allowed

to be between 0 and 10. The range for each method (METCOD=2 for BINSAM, NSMPLS between 0 and 10, and METCOD=5 for RANSAM, NSMPLS between 1 and 4) should be screened in INPM4.

A day is computed for each trial. The number of trials is 365 times the number of samples, so each day occurs NSMPLS times. Each day is partitioned according to the number of samples per day and an hour is selected from each partition using the random number generator, RANDOM.

In calculating the day and hour for the trial, the logic is more complicated than necessary and could be implemented to give a clearer picture of what is to be accomplished. A proposed change to RANSAM to help clarify this logic would be:

```

C  NSMPLS HAS A VALUE OF 1,2,3, OR 4
C  NUMTRI HAS A VALUE OF 365, 730, 1095, OR 1460
C  IPARLN HAS A VALUE OF 24,12,8, OR 6
C  IPARTI HAS A VALUE OF 0 WHEN IPARLN=24, 0, OR 1 WHEN IPARLN=12
C  0, 1, OR 2 WHEN IPARLN=8, AND 0, 1, 2, OR 3 WHEN IPARLN=6
    NUMTRI = NSMPLS*365
    IPARLN = 24/NSMPLS
    DO 120 ITRI = 1, NUMTRI
      IPARTI = MOD (ITRI-1, NSMPLS)
      CALL RANDOM
      JHOUR(ITRI) = RANVAL * (IPARLN-1) + 1 + IPARLN * IPARTI
      JDAY(ITRI) = 1 + (ITRI-1)/NSMPLS
120  CONTINUE.

```

Also note that PRBMET is constant and can be calculated before the loop at 900. As with the other routines in the weather sampling and binning, SNL has determined that all the routines need to be rewritten to clarify the logic and feel that changes such as this are not necessary in the current code.

The random number generator, RANDOM, was originally based on a linear congruential model modulo 256. There are exactly 256 different numbers "randomly" generated between 0.0 and 1.0.

In RANSAM, a sequence of 256 random integers was generated between 1 and 24/NSMPLS. Therefore, the sequence of hours repeats itself every 256 days

when NSMPLS=1, 128 days for NSMPLS=2, 85 and 1/3 days when NSMPLS=3 and every 64 days when NSMPLS=4.

In the BINSAM routine, there are currently up to 40 bins or categories and between 1 and 10 samples are allowed from each bin. Therefore, the total number of random numbers required is not large. With the unnecessary calculation following the final selection, there are at most 40 times 11 or 440 calls to RANDOM when using the weather category bin sampling. The question was whether some weather sequences may be specifically excluded using the random number generator which has a period of 256.

Anytime the number N/K is greater than the period of the random number generator, numbers will be excluded from sampling. In a worst case, with all 8760 hours in the same category, and sampling just once, the ONLY weather sequences which may be chosen are: $1 + \text{INT}(\text{RANVAL} * 8759) = 1, 35, 69, 104, \dots$

With K having values of 1 to 10, what is the minimum N must be before numbers are excluded? The following table gives these values:

<u>K</u>	<u>N</u>
1	256
2	512
3	768
4	1024
5	1280
6	1536
7	1792
8	2048
9	2304
10	2560

In the sample problem several weather categories exceed 256 values and two exceed 1024. Therefore

certain weather sequences can never be chosen as a trial, regardless of the seed for the random number generator.

The other concern regarding this random number generator was that there is a strong correlation between every other number in this sequence. This is not critical for MACCS because the number generated is used in a "coin-toss" fashion to select start times from the meteorological bins. It is recommended that a robust random number generator be chosen with a period of at least 8760, and preferably larger, on the contingency that some future version of the code would utilize multi-year weather data.

SNL agreed and rewrote RANDOM using a shuffled linear-congruential generator with an essentially infinite period because of the shuffling method utilized. This new random number generator was placed in MACCS prior to the NUREG-1150 revisions. The new routine was reviewed and found to be coded as documented by SNL.

The major conclusion from this review of the meteorological data routines is that they can all use substantial rewriting to improve understandability. SNL has agreed with this assessment and have indicated that they intend on doing so in the future.

2.6 Input Processing and Miscellaneous Routines

The following section covers the review of the balance of the MACCS 1.5 code. The routines discussed here are primarily those that are common to the previously reviewed program modules. These include the input routines that comprise the INPRE processor in MACCS 1.5 and any outstanding miscellaneous routines. The following contains a brief description of the INPRE processor with comments regarding potential errors or alternative programming. The final part of the section contains comments relative to any miscellaneous routines not reviewed in previous sections. Only those routines with potential errors or suggested programming improvements are discussed.

The INPRE processor in MACCS 1.5 is a free-field input processor. The processor was designed to allow comment cards to be freely interspersed within the input. Data required by MACCS 1.5 can be logical, character, integer or real. The INPRE processor determines the type and verifies that the type is correct. Subroutine RDSTRG converts an input record string to a character value, a logical value, a real value, or an integer value using the following logical functions. Logical function IMLGCL returns a TRUE if it detects a logical value, otherwise it returns FALSE. Logical function IMNTGR returns TRUE if it detects an integer value (with optional sign), otherwise it returns FALSE. Logical function IMDIGT returns TRUE if it detects an unsigned digit string, otherwise it returns FALSE. Logical function IMREAL returns TRUE if it detects a real value, otherwise it returns FALSE. RDSTRG assumes the input record is a character string if it is not a logical, real or integer.

Of these program units, IMLGCL, IMNTGR, and IMDIGT are short, however RDSTRG and IMREAL are quite long and are difficult to understand. IMREAL contained a potential bug. It involved a poor programming practice, and IMREAL got away with this only because RDSTRG (which calls IMREAL) contained some complex coding which compensated for it; however, if IMREAL was ever called from different circumstances, this could become a very real error.

The potential error in IMREAL occurs because if IMREAL is invoked with a signed value, IMREAL will shift the value of the argument to remove the leading sign. The concern occurs because IMREAL does not blank out the position of the last character (which is no longer used). For example, if the argument to IMREAL were -12.34, then IMREAL will shift it to

create 12.344. IMREAL also changes its internal length of the argument, so that it will not get into trouble, but it nevertheless has changed the value of the argument, and the changed value will be passed back to the invoking program unit. MACCS 1.5 gets away with this for now because RDSTRG places a copy of the argument in a temporary character string, TMPCRD, invokes IMREAL with the original character string, then decodes the information out of TMPCRD. This introduces unnecessary complexity in RDSTRG, and creates a potential error if IMREAL is invoked under different circumstances.

The suggested change to the code was to rewrite all of the INPRE processors to make them easier to read and understand. The primary change that needed to be made was the problem with IMREAL. SNL agreed with the fact that IMREAL was in error and revised the coding to copy the argument to a dummy variable before processing it. This change was made after the NUREG-1150 revisions. The other INPRE routines were not modified.

Following determination of the type of input record, MACCS 1.5 then reads the input. There are seven functions used to perform this operation. These functions and their purpose are:

1. FUNCTION CGET1 — Returns a single character value
2. FUNCTION IGET1 — Returns a single integer value
3. SUBROUTINE IGETN — Returns a sequence of integer values
4. FUNCTION LGET1 — Returns a single logical value
5. SUBROUTINE LGETN — Returns a sequence of logical values
6. FUNCTION RGET1 — Returns a single real value
7. SUBROUTINE RGETN — Returns a sequence of real values.

Each of the above return values of TRUE for the variables FOUND and RANGE if appropriate values are found and are within the specified range. This is good, and probably even justifies the changing the values of the function arguments, which should not usually be done in FORTRAN, because of increased complexity and error potential. There was however,

a basic problem which is common to all of these program units. The values of FOUND, RANGE, and the function values are not defined early enough in the program unit. Each of the above program units contain tests to ensure that they are being invoked under proper circumstances. If these tests failed, a message is printed and control passes back to the invoking program unit without FOUND, RANGE, or the function values being defined. This means, for example, that, IGET1 could be invoked successfully and would return a value of TRUE for both FOUND and RANGE, but at the next invocation, IGET1 could fail, print a message, and return with both of these variables still set to TRUE.

Another way of expressing the problem is to say that if these program units detect an input condition which causes them to abort processing, they will print a message for a human reader, but they may not pass any information back to the invoking program unit which indicates that they did not successfully process the data.

The solution is simple, in each of the above program units, the first executable statements should define FOUND and RANGE, presumably to FALSE. For the FUNCTION program units, the function values (CGET1, IGET1, LGET1, RGET1) should also be defined to default values. Currently FOUND is initialized in IGET1, LGET1, and RGET1, but it occurs after the initial input acceptability test. In IGET1 and RGET1, RANGE is also initialized, but to the value TRUE. It is not clear why RANGE should not be initialized to FALSE.

SNL determined that the basic error identified here was the violation of the ANSI standard to have an undefined function value as existed in CGET1, RGET1, LGET1, and IGET1. SNL changed these FUNCTION program units after the NUREG-1150 revisions. SNL determined that the definition of FOUND and RANGE would not affect program execution and opted not to change the routines at this time.

The balance of the coding consists of overall program control routines that set up and control the simulation and output routines that save and process CCDF summary information for printing. Additionally, the input and output routines associated with the ATMOS, EARLY, and CHRONC program modules were reviewed again relative to overall program logic and data flow. The following comments identify those routines in this group that were found to have errors or questionable programming structure. Unless stated otherwise, the changes identified were considered by SNL to be

candidates for an overall code cleanup at a future date as they do not affect program results and the proposed changes have not as yet been made to MACCS 1.5. Inconsistencies between MACCS 1.5 and the User's Guide variable limits were handled by SNL through changes in the User's Guide documentation.

RANSAM: performs stratified random sampling

The probability PRBMET is defined as a constant with the loop on 900. This could be moved outside the loop.

PRBTAB: This is an inactive routine consisting of 85 lines that have been commented out. As it is no longer used, it should be removed. SNL removed this routine after the NUREG-1150 revisions.

WRANBN: initializes the weather bin codes

Only one reference to DIRBI. Should use the array reference defining it instead and potentially save some time in the loop on 10.

PUTSTG: routine to load and fetch (entry GETSTG) the evacuation strategy input parameters.

The test to RETURN when NSTRTG=1 is redundant with the test in CONTRL.

PUTSTM: routine to store and fetch (entry GETSTM) source term data.

The number of source terms allowed (60) is used by local storage A1. This may complicate later modification of the code.

The data statement for ILNTHA could also be avoided if the size declarations for the common block MULREL are made part of a parameter statement and the size ILNTHA calculated therein.

The test to RETURN when NSRCTM=1 is redundant with the test in MACCS.

INPUT: routine controlling the processing of all user input for the MACCS modules

At line 57, the value of the maximum allowed number of source terms is hard-wired into the code. Should be parameterized elsewhere to allow flexibility in changing this limit.

When input processing changes to EARLY or CHRONC input, the value of OK was not

re-initialized. This may lead to misleading error messages.

INPBEG: routine to set up a database of user input data.

CLOC could be passed to the SEARCH routine as an actual argument.

INPEND: routine to close a user input file database.

The test for input data which exceeds the maximum length will miss any records which have a blank in the MAXLEN column and exceed that length.

SEARCH: locates the record CLOC using binary search.

This routine could be better organized. It was difficult to read in its current form.

SORT: sorts n values of character array card in increasing order.

This routine has not been well structured.

ERRFIL and ERRLOC: identify errors by writing a message to UNIT 6.

These two routines are identical in functionality and should be replaced with a single routine.

INPGEO: defines the geometric grid to be used.

Here and throughout the input processors, the logical variable OK is used to describe whether all previous input has been successfully read and stored in the variables to be used by the code. As described under the discussion on the INPRE processor, the core i/o routines return values of RANGE and FOUND to describe whether the variable was found in the input database and if the value is within the proscribed range. Recommend using the following construct:

OK = RANGE .AND. FOUND .AND. OK

instead of

IF (.NOT. RANGE .OR. .NOT. FOUND) OK = .FALSE.

In the instances where it is necessary to terminate input processing, then the test simply becomes:

IF (.NOT. OK) RETURN

INPISO: defines the isotope data used in the atmospheric model.

The upper limit on the variable NUCNAM was not consistent with the documentation: 10 vs 8

The upper limit on the variable NAME was not consistent with the documentation: 10 vs 8

The lower limit on the variable HAFLIF was not consistent with the documentation: 1.E-35 vs 1.

INPDIS: gets the dispersion parameter data

The lower limits on the variables CYSIGA, CYSIGB, CZSIGA and CZSIGB were inconsistent with the documentation: 1.E-35 vs 1.E-6

The lower and upper limits on the variables YSCALE and ZSCALE are inconsistent with the documentation: .01 vs .001 and 100. vs 1000.

INPREL: defines the release description of the plume.

The order of the variables read in this routine was not consistent with the order in which they are described in the User's Guide.

INPM6: Another inactive routine consisting of 163 lines that have been commented out. SNL removed this routine after the NUREG-1150 revisions.

INPOPT: defines the output options for the atmospheric model

The flag ENDAT1 was documented in a different section of the User's Guide. Move the segment of code to read and validate ENDAT1 to where read per documentation. SNL changed the documentation in the User's Guide to match the code.

INPMET: defines the characteristics of the weather sampling by reading from the problem input files.

An undocumented feature, METCOD=6, was not operational in MACCS 1.5. All references to the logical variable OK6 should be removed.

INPM1: loads the M1 common block data for weather sampling.

The test at line 49 should be for METCOD greater than 5, since METCOD=5 was the highest documented value. SNL removed the test from INPM1 after the NUREG-1150 revisions.

The call to WRDMET at line 57 may reset the value of OK from FALSE to TRUE.

WRDMET: reads the meteorological data.

Stability class 7 is allowed on input. This was not documented. SNL added required documentation to the User's Guide.

At line 87, MRN should be tested greater than (GT) 999, since 999 is allowed per the documentation.

INPM4: loads M4 common block for weather sampling.

Since INPM4 is called only with when METCOD has values of 2 or 5, the test for METCOD not equal 2 actually means METCOD equals 5.

The documentation says that two to four rain intensities may be specified, with one to three rain intensity break points defined by a variable named NRINTB, which was never referenced in the code. Instead, the variable NRINTN was read with a range allowed between two and three. This variable is later incremented by one, implying a range of three to four rain intensities. SNL changed the variable name in the documentation to NRINTN. Documentation was added to the Model Description to describe the weather binning procedure in detail.

The several transfers to statement labelled 310, erroneously bypass the statement at line 204 incrementing NRINTN. This could result in improper diagnostic error messages for the subsequent variables if an invalid input value is supplied.

WBNMET: determines groupings for one years meteorological data.

Type and size declarations should be combined for easier modifications. The number of wind directions should be placed in a PARAMETER statement:

PARAMETER (ISMW=16, ISMRY=ISMW+1)

The value of KRAIN is restricted to be less than or equal to 999 in WRDMET. The use of the statement function MRAIN seems pointless here.

INPM5: loads M5 common block for weather sampling.

The lower bound of 100 for HRMXHT was inconsistent with the documented lower bound of 0.

INMISC: defines the miscellaneous information for the run.

The loop on 120 should define the constant 1./NUMCOR outside of the loop and not compute it for each component.

The upper bound of 10 for IPRINT was inconsistent with the documented value of 8.

EDCINP: reads the "EARLY" dose conversion factors.

There is similar logic in this routine and elsewhere when checking an input name (organ, nuclide, etc.) against an existing list of acceptable values. There is a potential here to reduce the size of the code by building a single common routine.

INEVAC: reads the evacuation network data for the early health effects model.

The upper bound of 6 for WTNAME was inconsistent with the documented value of 7.

The upper bound of 86400. for EDELAY was inconsistent with the documented value of 1.E6.

The variable ESPEED should be read in this routine rather than in the subordinate routine EVRADI. SNL disagrees with this recommendation.

INPOPU: defines the population distribution surrounding the site.

The input range check on POPFLG allowed a character string of length 4 to 7. It may have a value other than "UNIFORM" or "FILE" and pass the RANGE and FOUND check. This logic should be combined with the following IF-THEN-ELSEIF block testing for those two values.

The upper bound of 1.E6 for POPDEN was inconsistent with the documented value of 1.E10.

A single error check following the input of POPDEN and IGENIN would suffice to terminate input if the population distribution cannot be calculated.

POPDEN is scaled by 1.E-6 at line 134. Appears to be a conversion from people/km² to people/m² although there are no comments in the code to this effect.

CMPTBL: checks that NINPT and NMODL are identical.

The logical variable OK was not initialized to TRUE. A potential problem exists if this argument is not initialized in the calling routine.

MATCH: checks that SEPRTR and SDFKEY match

The order of the arguments does not parallel the order in CMPTBL, a similar routine.

The logical variable OK was not initialized to TRUE.

EVRADI: reads radial evacuation data from the input database and creates an evacuation network to represent it.

Move the read of the variable ESPEED to INEVAC. SNL disagrees with this recommendation.

EVNETW: reads evacuation network data from the input database.

This routine is never called since the value of IEVACU in the parent routine INEVAC is set to 1 meaning only radial evacuation is allowed. No further analysis of this routine was performed.

EVROOT: makes a list of root nodes within the evacuation zone.

Connected with the network evacuation performed by EVNETW and, therefore, never called in the current MACCS 1.5 code configuration. No further analysis of this routine was performed.

INPEMR: defines the emergency response zone model.

The upper bound of 86400 for both variables TTOSH2 and SHEL2 were inconsistent with the documented values of 1.E7.

The lower bound of 2 characters for the variable CRIORG was inconsistent with the documented value of 3.

INEFAT: defines the early fatality risk model parameters.

The lower bound of 2 characters for the variable NAME was inconsistent with the documented value of 3.

INEINJ: defines the early injury models.

The lower bound of 2 characters for the variable NAME was inconsistent with the documented value of 3.

INACAN: defines the model for risk of cancer from acute exposure.

The lower bound of 2 characters for the variable NAME was inconsistent with the documented value of 3.

INOUT3: defines the options for result number 3.

The lower bound of 2 characters for the variable NAME was inconsistent with the documented value of 3.

INOUT4: defines the options for result number 4.

The IF-THEN-ELSE block testing NAME is very similar logic to the block in INOUT1. Another opportunity to simplify some of the coding.

INOUT5: defines the options for result number 5.

The lower bound of 2 characters for the variable NAME was inconsistent with the documented value of 3.

The loop at 300 is an example of a WHILE construct simulated using a FORTRAN DO loop. The logic would be more easily understood if we replace

```

DO 300 IORG = 1, NUMORG
IF (NAME .EQ ORGNAM(IORG)) THEN
  INDEX5(IOUT) = IORG
  GO TO 300
ENDIF
300 CONTINUE
OK = .FALSE.
WRITE(6,*) ' ORGAN NAME NOT FOUND
CALL ERRLOC ('INOUT5','ORGNAM)

```

with

```

IORG = 1
300 CONTINUE
IF (IORG .LE. NUMORG) THEN
  IF (NAME .NE. ORGNAM(IORG) THEN
    IORG = IORG + 1
    GO TO 300
  ELSE
    INDEX5(IOUT) = IORG
  END IF
ELSE
  OK = .FALSE.
  WRITE(6,*)...
  CALL ERRLOC...
END IF

```

The lower bound of 1 for the variable I2DIS5 was inconsistent with the documented value of I1DIS5.

INOUT6: defines the options for result number 6.

The lower bound of 2 characters for the variable NAME is inconsistent with the documented value of 3.

The lower bound of 1 for the variable I2DIS6 was inconsistent with the documented value of I1DIS6.

INOUT7: defines the options for result number 7.

The first test in the routine should be on the value of IPLUME, which occurs after the first read from the input database.

The IF-THEN-ELSE block testing NAME is very similar logic to the block in INOUT1 and INOUT4. Another opportunity to simplify some of the coding.

The lower bound of 1 for the variable I2DIS7 is inconsistent with the documented value of I1DIS7.

The final logic starting at line 198 could be reversed for clarity to include the loop 800 inside an IF block when OK is TRUE.

INOUT8: defines the options for result number 8.

The IF-THEN-ELSE block testing NAME is very similar logic to the block in INOUT1, INOUT4, and INOUT7.

The final logic starting at line 190 could be reversed for clarity to include the loop 600 inside an IF block when OK is TRUE.

OPNERL: copies modelling data from the common blocks used by EARLY into the common blocks used by CHRONC

UNFSWT is undefined if POPFLG has a value other than the two allowed.

INCHRN: defines the user input for the CHRONC code.

The lower bound of 1.E-20 for variable DSCRLT was inconsistent with the documented value of 1.E-35.

The upper bound of 8 characters for the variable CRTOCR is inconsistent with the documented value of 10.

The upper bound of 1.E5 for the array CDNFRM was inconsistent with the documented value of 1.E35.

The conversion factor from years to seconds should also be in a PARAMETER statement. Note that elsewhere in the code a year was assumed to be 365 days, whereas in this routine the year was 365.25 days. SNL states that this discrepancy has been corrected and 365.25 is now used throughout the code.

STPATH: reads and initializes the chronic ingestion pathway data.

The complexity of this routine seems to be due to the fact that it performs in reality nine sequential functions. Recommend that this routine be reorganized in a more modular design, separating the functions.

The conversion factor from years to seconds should be in a PARAMETER statement.

IXOT9: defines the options for result type 9.

The lower bound of 2 characters for the variable NAME was inconsistent with the documented value of 3.

The lower bound of 1 for the array IX2DS9 was inconsistent with the documented value of IX1DS9.

IXOT10: defines the options for result type 10.

The lower bound of 1 for the array I2DS10 is inconsistent with the documented value of I1DS10.

IXOT12: defines the options for result type 12.

The lower bound of 1 for the array I2DS12 was inconsistent with the documented value of I1DS12.

STTIMS: sets times for the chronic and economic models.

The conversion factor from years to seconds should be in a PARAMETER statement.

SDFINP: reads and checks the site data file.

Although this unit was previously read by INPOPU, there is no provision for error processing on the first read in the event the file is empty or nonexistent. SNL states that this test would be unnecessary. SNL has incorporated explicit OPEN and CLOSE statements on all files used by MACCS 1.5 which avoids problems encountered with empty input and data files.

There is no call to ERRFIL in the IF block beginning at line 199.

CXPTBL: check that NINPT and NMODL are identical.

The logical variable OK was not initialized to TRUE. Potential problems exist if this argument is not initialized in the calling routine.

MXTCH: checks that SEPRTR and SDFKEY match

The order of the arguments does not parallel the order in CXPTBL, a similar routine.

The logical variable OK was not initialized to TRUE.

EXCINP: reads the dose conversion factors.

The read statements to skip the 2 line intermediate header should be juxtaposed for clarity instead of separated by 130 lines of coding.

HEDEAR: prepares the result list for the EARLY module.

The maximum number of results, MAXNRS, should be in a PARAMETER statement.

The names of the results categories should be in DATA statements rather than assignment statements from the standpoint of efficiency.

The WRITE statement at line 181 could be moved to line 219 to consolidate the output when STOPIT is TRUE.

COPCHR: sets up the CHRONC code to produce the results that are requested by both EARLY and CHRONC.

Results of type 1, 4, 7, and 8 are keyed with codes between 100 and 300. A different way of flagging this could have been used to be more intuitive.

HEDCHR: loads the data necessary for CHRONC to generate results.

The WRITE statement at line 149 could be moved to line 171 to consolidate the output when STOPIT is TRUE.

SNL discovered additional errors in these routines after the line-by-line QA and verification and before the release of MACCS 1.5. SNL went through MACCS 1.5 and made necessary changes so that all occurrences of NUCNAM have a length between 3 and 8 characters and all occurrences of ORGNAM have a length between 2 and 10 characters. The documentation in the User's Guide was made consistent with these changes, thereby removing the previously identified discrepancies in the bounds of variables NAME and NUCNAM. This was done after the NUREG-1150 revisions. SNL discovered a typographical error in READ2 where the argument list in the call to DO1CDF should have a zero as the first argument for "fraction of the time" weighting in overall combined results. SNL also discovered an error in READ2 that produced erroneous overall combined results if CHRONC is being skipped and more than one emergency response is being used. The error caused the last emergency response to be double-counted. These errors in READ2 were corrected prior to the NUREG-1150 revisions. SNL added logic to INPBEG to protect the user from code failures resulting from adding new data to multiple run change cards after the NUREG-1150 revisions. SNL added logic to all of the output routines in EARLY and CHRONC to allow generation of CCDF tables in the output after the NUREG-1150 revisions.

3. MACCS 1.5 DOCUMENTATION REVIEW

The MACCS 1.5 code draft documentation provided by SNL was reviewed as part of the QA and verification effort. The documentation is composed of three volumes. Volume I, the User's Guide, describes the input data requirements of MACCS 1.5 and provides directions for using the code. Volume II, the Model Description, describes the underlying models that are implemented in MACCS 1.5. Volume III, the Programmer's Reference Manual, describes the MACCS 1.5 code structure and database management. The following sections describe each of these volumes relative to their completeness and accuracy.

3.1 User's Guide

The MACCS 1.5 User's Guide reviewed here was a draft issued by SNL on May 11, 1989. A previous draft version of this manual correcting errors noted during the line-by-line QA of MACCS 1.5 had been reviewed and the comments provided to SNL. The May 11 draft incorporated those comments and it is that draft, referred to hereafter as the draft User's Guide, which is discussed here. The User's Guide provides the directions for the preparation of an input deck and procedures for execution of a MACCS 1.5 simulation. The first error noted in this manual occurs in the abstract where it stated that there are only two volumes of documentation. Volume III, the Programmer's Reference Manual, is not mentioned and should be. The User's Guide begins with an overview of MACCS 1.5 from the perspective of the code user. This is followed by descriptions of the data required for the ATMOS, EARLY, and CHRONC program calculations. Appendices are included to provide variable lists, auxiliary data files format, procedure files to run MACCS 1.5, program structure diagrams, and input files for the three sample problems provided on the MACCS 1.5 transmittal tape.

The overview describes the objectives of MACCS 1.5, the structure of the code, the user input processing, and a description of the supplied sample problems. This section is very well written and provides an excellent overview of the code. The user is presented with various operational modes of MACCS 1.5 and directed to appropriate sections of the User's Guide for additional information. The sample problems are discussed relative to MACCS 1.5 code features exercised and types of calculations being performed. The overview is explicit in its declaration that parameter values shown throughout the guide and in the sample problems are not to be construed as recommended values. The

overview directs the user to a data review document^a which describes appropriate calculations performed to generate the MACCS 1.5 input decks used for the NUREG-1150 revisions. This document was under development at this writing and was not available for review.

The next three sections of the User's Guide are devoted to the ATMOS, EARLY, and CHRONC program module input files, respectively. These sections describe the data required by the various program modules of MACCS 1.5. The data are divided into data groups identified by a two character mnemonic. A short description is provided for each data group as to its purpose and use within the program. Those data blocks that can be changed in subsequent change cases (ATMOS and EARLY) are clearly identified. The input data descriptions within each data group include the name, type, allowed range, purpose, and example usage of each variable. The review ascertained that the name, type, range and purpose in the documentation matched that implemented in MACCS 1.5 with one exception. The allowed number of watersheds was increased from two to four after preparation of the draft User's Guide so the CHRONC ingestion transfer data (Section 3.10) did not reflect this change.

The example usage given for each variable was stated in the overview not to be taken as a recommended value for that variable. There are no references given in the draft User's Guide for these example usage values unlike the earlier versions of the User's Guide. SNL states in the overview that the user must ensure the appropriateness of all data in the input. The data given as example usage are taken from the sample problem input decks and are for the most part identical to those data used for the NUREG-1150 revision calculations. By removing references for variable example usage, the user must go to the references found in the Model Description and the NUREG-1150 revisions data review document to determine justifiable values and calculational methodology for each variable. This is an acceptable method provided that the reference material is readily available and contains information commensurate with the MACCS 1.5 input requirements. Of these two references, only the draft Model Description was available for review and

a. J. L. Sprung et al., *Evaluation of Severe Accident Risks: Vol. 2, Quantification of Major Input Parameters: Part VII, MACCS Input, Draft Revision 1*, NUREG/CR-4551, SAND82-0342, Sandia National Laboratories, to be published.

therefore no assessment can be currently made relative to this requirement.

The draft User's Guide contained numerous instances where model descriptions are interspersed with the input descriptions. This information is also found in the draft Model Description and is merely duplicated in the User's Guide. In some cases, the duplication serves the purpose of restating code limitations or interactions between the three program modules. In some cases, the model description contained in the User's Guide was either clearer than or not found in the draft Model Description. In most cases, however, the model description information should be removed and reference made to the appropriate section of the Model Description. The description in the Model Description should be as clear and complete as that currently contained in the draft User's Guide. In ATMOS, the discussion of the meteorological sampling scheme in Sections 1.13 and 1.16 could be replaced. In EARLY, the early fatality, early injury, and latent cancer data discussions in Sections 2.8, 2.9, and 2.10, respectively, could also be replaced. In CHRONC, the discussions concerning decontamination plan, groundshine weathering, resuspension weathering, ingestion transfer factors (primarily the washoff and crop transfer discussions) data contained in Sections 3.5, 3.7, 3.8, and 3.10, respectively could also be replaced. The inclusion of this information does not degrade the readability of the User's Guide but is simply redundant with that contained in the Model Description.

There were five Appendices to the User's Guide. These were: a glossary of all input variables (Appendix A), auxiliary input files (Appendix B), procedure files to run MACCS (Appendix C), program structure diagrams (Appendix D), and user input files for the

three sample problems (Appendix E). The glossary in Appendix A provided a very useful alphabetical listing of all input variables, their associated program module, the section of the User's Guide where the variable was defined, and a brief description of the variable.

Appendix B contained format information for the auxiliary files used by MACCS 1.5. The information includes variable units, format, type, name, description, allowable range, and, where applicable, the variable in the input deck that the auxiliary file variable is compared to. The files described are the meteorological data, dose conversion factors, and site data files. In the description of the meteorological data file, the description indicates a range of -1 to 999 100ths of in. for accumulated precipitation. As discussed in Section 2.5 of this report, MACCS 1.5 currently converts the accumulated precipitation to tenths of mm/hr before testing and thus limits the maximum allowed rate to 3.93 in./hr, or a maximum range of 393. This needs to be either changed in the code or correctly stated here. In the discussion concerning the dose conversion factors, the organs EDEWBODY and THYROIDH were not listed even though they are included on the current dose conversion factors file listed in Appendix E. EDEWBODY is also mentioned in the main body of the User's Guide (Section 2.4) as being included in this file. In the site data format, the maximum number of watersheds needed to be increased from 2 to 4 and the maximum number of economic regions needed to be increased from 59 to 99. The increase in the maximum number of watersheds also needs to be incorporated in the presentation of and format for the watershed data block. The presentation on page B-12 needs to have an additional zero representing the ingestion factor for the additional watersheds for each isotope shown. On the same page, the format block needs to include the following entries:

<u>Columns</u>	<u>Format</u>	<u>Variable</u>	<u>Identification</u>	<u>Allowed Range</u>
56-65	E10.1	WTRINF(3)	Ingestion factor for watershed class 3	0.0-1.0
66-75	E10.1	WTRINF(4)	Ingestion factor for watershed class 4	0.0-1.0

Appendix C discusses the procedure file for executing MACCS on the SNL VAX/VMS. It is intended to be used as a guide for running MACCS on VAX/VMS computer systems. The only omission noted was that the write-up doesn't state that the command file MACCS.COM discussed here is included on the distri-

bution tape. The only error noted was the reference to MACCS 1.6 should be MACCS 1.5.

Appendix D contained program structure diagrams of MACCS 1.5. These diagrams are also contained in the MACCS 1.5 Programmer's Reference Manual and

should not be included here. It is recommended that this Appendix be removed from the User's Guide. SNL states that this has been done.

Appendix E contained listings of the three sample problems and the auxiliary data files transmitted with MACCS 1.5. SNL again states in the preface to Appendix E that the values shown "should not be construed as recommended parameter values for the assessment of reactor accidents." The only item noted in this Appendix was the inclusion of organs EDEWBODY and THYROIDH in the dose conversion data listing that were not discussed in Appendix B.

3.2 Model Description

The Model Description is Volume II of the MACCS 1.5 documentation. It describes the underlying models that are implemented in MACCS 1.5. The July 15, 1989 draft version of the Model Description was reviewed and is discussed below. No final draft was received prior to the writing of this report. The following summarizes the findings of that review which were provided to SNL for incorporation into the final documentation. The comments only cover items that are errors in the documentation relative to the code or that were required to improve understanding of the models being described. Editorial comments are not described here.

The MACCS 1.5 Model Description is a substantial improvement over the MACCS 1.4 Reference Manual used for the line-by-line code QA and verification. The model discussions are much more detailed in this draft version and the references are more complete. As with the MACCS 1.4 Reference Manual, the Model Description is not organized into the ATMOS, EARLY, and CHRONC sections found in the User's Guide. The Model Description describes the MACCS 1.5 code models in sections discussing atmospheric transport, dosimetry and exposure pathways, economic effects, mitigative actions and dose accumulation, and health effect models. The first section describes all the models contained in the ATMOS program module but the models that make up the EARLY and CHRONC program modules are interspersed throughout the rest of the manual. The Model Description subdivides each of these sections into the early or emergency phase followed by the long-term or chronic phase. Although a bit awkward at first, this approach does group similar models together and is probably the most efficient way to present the information. Finally, the Model Description contained four Appendices that provide a list of data that are hard-wired into MACCS

1.5 (Appendix A), a description of the weather bin sampling method (Appendix B), the methodology for calculating the maximum allowable ground concentrations for food pathway isotopes by a preprocessor program (Appendix C), and the MACCS 1.5 dose conversion factors (Appendix D).

Section 1, the introduction to the Model Description, was found to contain too much detail. This section should provide a general overview of the code relative to subsequent sections of the manual. There were several instances of equations and figures that were felt to be inappropriate in the introduction. The equation given for the risk of an early health effect, Equation 1.2.1, was in error with the minus sign and the summation sign transposed. A few paragraphs were not clear in the presentation and a rewrite was requested to clarify them.

Section 2 describes the atmospheric transport models used by MACCS 1.5 to calculate the downwind transport, dispersion, and deposition of radioactive materials. The section is very detailed in its treatment of these models. The author did tend to mix units, i.e., miles and kilometers, which should all be in SI metric units. The equations for critical windspeed (Equation 2.8.1) and both stable (Equation 2.8.2) and unstable (Equation 2.8.3) plume rise were missing the scaling factors that are in the code implementation. The model for windspeeds aloft (Equation 2.8.5) failed to mention the MACCS 1.5 imposed limit of 200. m. in the windspeed height. In the discussion of the Gaussian plume models, there was a typographical error in the subscript for crosswind standard deviation for the unconstrained time-integrated air concentration (Equation 2.9.1). The Gaussian equation that accounted for the effect of the ground and the capping inversion layer (Equation 2.9.2) was missing the term for a nonzero crosswind distance. There was no discussion given for the average plume size across an interval in which the stability class changes. The wet deposition model description was found to be inconsistent relative to Figure 2.3 in reference to the timing of the plume head and tail. The average centerline air concentration (Equation 2.11.5) incorrectly specifies $z=0$ instead of $z=H$. Finally, the references Ch89, Do79, Gi76, and Ka82 given at the end of the section were not referred to in this section.

Section 3 of the Model Description discusses the models used to calculate radiation doses for both the early and chronic phases of the exposure. The models are described for all cohorts (direct, indirect and decontamination workers) and for all pathways appropriate for the accident phase. Section 3 only presented the

models used for dose calculations for the two phases and did not address dose accumulation.

The early exposure portion of Section 3 contained no obvious errors but numerous instances where additional information was required to clarify the model description. The variable for the time-integrated air concentration, AC_i , used in the development of the cloudshine dose (Equation 3.1.1), inhalation dose (Equation 3.1.9), and skin surface contamination (Equation 3.1.13) didn't refer the reader back to Section 2, Equation 2.11.3, where it is developed with a different variable name. Reference to this equation with a discussion of its integration time would make the explanation of the ratio of exposure time and plume overhead duration in these equations less ambiguous. The discussion of cloudshine correction factors shown in Table 3.1 needs to state that the factor is equal to the table endpoint values when the table values are exceeded. The discussion of groundshine dose during (Equation 3.1.2) and following (Equations 3.1.4 and 3.1.8) plume passage used the same variable name for both doses. The actual groundshine dose is the sum of these two doses and should be specified as such. The discussion of skin dose contained numerous references to "exposed area" but never defines how this area is calculated.

The long-term exposure portion of Section 3 contained some errors as well as places where additional information was required for clarification. Equation 3.2.8, the food ingestion dose, should have FA instead of FAC for fraction of area devoted to farming and FAC_j instead of FC_j for fraction of farmland devoted to crop j . The dose in this equation also needs to be specified as an indirect population cohort dose. The long-term washoff model, Equation 3.2.19, should use $x_i(t)$ instead of x_i . The equations for $WSHTOT_i$ and $WSHFRC_i$ on page 3-28 should use x_{oi} instead of x_o . The discussion of off-centerline doses on page 3-29 was not clear relative to how the step size was determined in the crosswind histogram. The number of histogram steps that the user can specify incorrectly includes the value 1 on page 3-30. Finally, the references He85 and Ko81 listed at the end of the section are never used in the write-up.

Section 4 of the Model Description discusses the economic effects models in MACCS 1.5. The cost of loss of usage of nonfarm property, Equation 4.3.3, indicates that $DPRATE$ and $DSRATE$ were input by the user as sec^{-1} whereas they are actually input as year^{-1} and converted by the code to sec^{-1} . The costs $POPCST$ and $VALWNF$ in this equation are user supplied as \$/person instead of \$. These two errors occurred in the

cost of loss of usage of farmland, Equation 4.3.5, with the exception that costs are input as \$/hectare instead of \$.

Section 5 of the Model Description describes mitigative actions and dose accumulation. This section contained discussions that were very difficult to follow and that need some major rewriting to make them more intelligible. In the write-up on the emergency phase, variables referred to as shielding factors in the draft User's Guide were referred to as protection factors. The term protection factor is defined the last time it occurs in the section on page 5-6. The evacuation section on page 5-4 stated that all people within a designated region evacuate and failed to account for a "fraction of the people" less than 1. The write-up on the intermediate phase never clearly defined what the purpose of this phase is in a calculation. It is not used in any of the sample problems and either needs to be justified or removed from the code and the manual.

The long-term phase portion of Section 5 contains descriptions found to be overly difficult to follow. Section 5.3.1.1 on decontamination and interdiction on page 5-12 was written as a series of steps with nested IF tests. The FORTRAN in MACCS 1.5 was actually easier to understand than this model description. The user supplied value of the decontamination factor, $DECON$, never described whether it includes radioactive decay or just material removal. Under dose accumulation on page 5-13, the Model Description incorrectly states that dose is accumulated from the time people return following decontamination. MACCS 1.5 always started at the time of deposition of material and multiplies by $DECON$ to reduce this dose. This needs to be changed in the write-up. SNL has rewritten the decontamination model in the code and has stated that the revised writeup in the Model Description accurately describes it. The calculation of $MINYRS$ described under restricted farmland production on page 5-16 does not say what assumptions are made for RS in Equation 5.3.3. Section 5.3.3.2 on cost effectiveness was also written as a series of steps with IF tests. The write-up should be expanded and converted to descriptive text to provide a clear discussion of this model. The discussion on growing season milk disposal, page 5-17, stated that disposal occurs for one-fourth of the growing when in fact the code used one-half. Finally, references A178, US77, and US82 were listed but never referred to in this section.

Section 6 describes the health effects models in MACCS 1.5. The first part of the section discusses early health effects models. Section 6.1.3 discusses the effective inhalation dose conversion factors but did not clearly state what the time periods are and where they

are set. In the same section, the discussion on dose protraction needed to be expanded to relate which output dose values are protracted and which are not. In Section 6.1.5 on page 6-6, the write-up claims that the cumulative hazard function for early fatalities given by Equation 6.1.9 can either overpredict or underpredict risks. The write-up needs to be more specific relative to the net effect on a MACCS 1.5 calculation.

The chronic health effects models are discussed in the last part of Section 6. In Section 6.2, page 6-8, the chronic cancer risk given in Equation 6.2.1c is a linear equation. In Section 6.2.1, page 6-9, the equation given with Table 6.2 indicates the chronic cancer risk is a quadratic. There are no references given for statements regarding high LET radiation effects on the dose-response models (page 6-9) or for the assumptions regarding "other" cancers in the discussion of Table 6.2 on page 6-10. Finally, reference Ch89 needs to be added to the reference list at the end of the section.

Appendix A of the Model Description lists the variables that are hard-wired into MACCS 1.5. The only items discovered missing were the fixed time periods used in the health effects models and the dose projection model.

Appendix B describes the weather bin sampling method used by MACCS 1.5. No reference was given for the selection of the sixteen initial condition bins discussed on page B-2 that are hard-wired into MACCS 1.5. The technique used by MACCS 1.5 to randomly draw sequences (page B-4) was also not described.

Appendix C provides the methodology for calculating the maximum allowable ground concentration PSCMLK, PSCOTH, and GMAXR. The methodology is contained in a preprocessor program, MAXGC, developed by SNL for calculating these input variables. The Appendix does not contain a listing of or documentation for the preprocessor. Equation C.1 is the same equation given in Section 3 (Equation 3.2.8) but uses different variable names. The discussion on the current growing season transfer factor CTD_{ij} on page C-18 introduces the term "weathering pattern" but does not define it. The Appendix leaves one wondering why the user must go through this process to back out a ground concentration when it makes more sense to have the MACCS code calculate it. It seems that a change to MACCS 1.5 is preferable to this procedure with its high potential for error. As a minimum, the preprocessor and appropriate documentation should be provided so that the user can reliably compute these variables.

Appendix D discusses the MACCS 1.5 dose conversion factors. The Appendix does provide some information on the methodology using the preprocessor, DOSFAC, to convert the ORNL data into MACCS format. However, it did not provide a listing of the program or any details about its availability. This prevents the user from generating dose factors for organs or isotopes different from those supplied with the code. This information should be provided in the introduction to this Appendix and the preprocessor provided with the code. Again, there are no references given for the multipliers used for high LET radiation dose conversions.

3.3 Programmer's Reference Manual

The Programmer's Reference Manual is Volume III of the MACCS 1.5 documentation. It describes the MACCS 1.5 code structure and database management. The August 31, 1989 draft version of the Programmer's Reference Manual was reviewed and is discussed below. No final draft was received prior to the writing of this report. The following summarizes the findings of that review which were provided to SNL for incorporation into the final documentation. The comments only cover items that are errors in the documentation relative to the code or that were required to improve understanding of the write-up. Editorial comments are not described here.

The MACCS 1.5 Programmer's Reference Manual is the first manual of its type for the MACCS code. Previously, Appendices to the User's Guide and internal code documentation were relied upon to provide programming assistance. The Programmer's Reference Manual provides a programmer's overview, a section on the MACCS subprograms, and a discussion of the MACCS data structures. An Appendix is provided that gives the individualized subprogram calling structure in MACCS 1.5.

Section 1, the programmer's overview of MACCS 1.5, provides general information about the MACCS family of computer codes. The introduction on page 1-1 does not provide any history of MACCS relative to publicly available versions or versions used for NUREG-1150. This should be included as the structure of MACCS 1.5 is significantly different than all previous versions. Section 1.4 provides a graphical representation of the hierarchical control structure of MACCS and Section 1.5 provides a list of the subprograms in the order in which they appear in the code. Both of these are reproductions of comment blocks contained in the MACCS FORTRAN. There were a few discrepancies that were primarily reflected current

changes to the code not yet reflected in the chart or list. A sequential flow diagram showing internal looping structure and a data flow diagram showing inter-module data migration are also provided in Sections 1.6 and 1.7, respectively.

Section 2, the MACCS subprograms, provides a description of the MACCS 1.5 subprograms and their interactions. Section 2.2 provides a calling structure tree that gives the order and number of calls within MACCS 1.5. The calls from WBNMET shown on page 2-6 were left off the chart. All occurrences of DOCCDF need to indicate that the balance of the called routines are found in the "frequently used routines" list. The routines called by WGTMET were not shown on pages 2-22 and 2-23. VELADJ is not called by PLMRIS as shown on page 2-27. INPGRC and DECAY were not called by CHROUT as shown on page 2-28. It was noted that named statement functions were not included in these calling charts.

Section 2.3 provides detailed subprogram descriptions for each subprogram in MACCS 1.5. The description includes the subprogram name, type, program module that it's associated with, purpose, subprograms that call it, and subprograms it calls. ABORT on page 2-32 was not called by CHROUT but is also called by OUTPUT, OXTPT4, OXTPT6, and OXTPT7. CCDOUT, on page 2-35, is no longer used by MACCS but is still in the code. CONTRL, on page 2-38, is also associated with the EARLY, CHRONC, and OUTPUT modules. DOCCDF, on page 2-41, is associated with the EARLY and CHRONC modules. ERRLOC, on page 2-45, is also called by INOUT2. GNBIN2, on page 2-49, no longer calls CCDOUT. INEFAT, on page 2-53, also unconditionally calls IGET1. INEVAC, on page 2-54, also unconditionally calls IGETN and RGETN. INOUT8, on page 2-59, also conditionally calls DOCCDF. INPM2, on page 2-63, also unconditionally calls RGET1. OUTPT1, on page 2-75, is called by STOEAR. IGET1, on page 2-97, is also called by IGETN. LGET1, on page 2-98, is also called by LGETN. As with the chart in section 2.2, named function statements are not listed in the call lists.

Section 3 provides the MACCS 1.5 data structures which include argument lists on external references, common blocks, and binary sequential files. The argument lists are said to be well documented in the code with variable descriptions given. The introduction on page 3-1 stated that there is only one hidden EQUIVALENCE statements in common blocks when, in fact, there are nine. These were in common blocks IRAIN (the one identified in the introduction),

CDATE, DCFCTR, DRFCTR, KPRINT, M2, NETWOR, REUSE1, and REUSE2. Section 3.2 lists all the common blocks in MACCS 1.5 with lists of the variables and arrays it contains and the subprogram units which use it. The unreferenced common blocks identified in Table 2 of this report showed up on this list as referenced as the list was compiled before they were removed. These will not be repeated here. There were numerous typographical errors that were provided to SNL that were not mentioned here. ACANCR, on page 3-3, also contained arrays DOSEFA, DOSEFB, CFRISK, AND CIRISK. CDATE, on page 3-4, was also used by WINCTM. CHNAME, on page 3-5, was also used by INCHRN. DRYCON, on page 3-8, contained array DSPCRP, not DSPOTH. HEADER, on page 3-13, was also used by STOEAR. IFF, on page 3-14, was also used by RANDOM. IUNIT, on page 3-16, contained array IUNIT.

KPRINT, on page 3-17, apparently had variable KLTPNT mistyped in subprogram CSTDCN as KTLPTN, where it was unused. LTACTN, on page 3-18, was not used by LTPROJ but was used by LTMACT. M2, on page 3-19, apparently had variable IBSTB mistyped in subprogram CONMET as IDBSTB, where it was used but not defined. This could cause a problem in the code. METB, on page 3-20, contained variable NTOT which was undefined and unused. METDAT, on page 3-20, was also used by USRSUP. METDTA, on page 3-20, contained variables MONTHS and ROSE which were undefined and unused. NAMCRP, on page 3-20, was also used by DIRDEP. NETWOR, on page 3-21, apparently had variable LASEVA mistyped in subprogram OPNERL to LXSEVA, where it was unused. REUSE1, on page 3-26, was also used by CCDOUT. WETCON, on page 3-30, was also used by WASHOU. WTRAC, on page 3-30, was also used by READ2.

Section 3.3 provides the variable trail for each common block variable and array in MACCS 1.5. The information provided for each variable and array is the common block name, subprogram using it, and whether the value is modified by assignment or argument or left unchanged. There were numerous errors in the subprogram usage list and variable modification as well as typographical errors in this section. These are too numerous to list here and were provided to SNL for correction in the final document. Most of the errors noted in this section were related to those identified in Section 3.2 of the Programmer's Reference Manual.

Appendix A contained detailed individual calling charts for any subprogram that call other subprograms

during execution. The charts used a series of numbers and letters to designate the subprogram calling level with 15 separate symbols in three chart structures to represent the IF, DO, and GO TO loops. The charts did not provide any line numbers or statement labels to orient the reader with the subprogram being represented. A user who desires to understand the loop

structure of any subprogram would find it easier to evaluate the FORTRAN than decipher these charts. The review determined that this Appendix added nothing to the Programmer's Reference Manual and should be removed. As a consequence, the charts were not reviewed in any detail relative to the MACCS 1.5 FORTRAN.

4. SUPPORTING REVIEW AND ANALYSES

The following summarizes the results of any reviews or analyses that are related to the QA and verification of MACCS 1.5. At the time of this writing, a review of the chronic exposure pathway models which had been completed and documented and a series of benchmark analyses which are currently planned were included in this category. The review of the chronic exposure pathway models was performed by Ulf Tventen of the Institute for Energy Technology in Norway under contract to SNL and is documented in Reference 15. The results of this review are discussed here. The benchmark analyses, utilizing the CSNI standard problems used for the international consequence code comparison study, are to be completed by the NRC research staff and documented as a NUREG.^a These analyses are designed to compare MACCS 1.5 results to those obtained from the international code comparison study. As the calculations have not yet been completed, the results of this comparison were not available for this report. They are mentioned here to provide the reference for this important piece of work for the reader.

The review of the chronic exposure pathway models in MACCS 1.5 was completed using the same code versions and documentation used for the line-by-line verification documented in this report. Additionally, the review relied heavily upon the January 1988 draft copy of NUREG/CR-4184, Ostmeier's unpublished report on exposure pathway models. The review included examination of the MACCS 1.5 implementation of the chronic exposure pathway models and a comparison of these to the models found in similar codes in use by OECD member countries. These codes include ARANO (Finland), CRAC/CRAC2 (USA), NECTAR (United Kingdom), NUCRAC (USA), and UFOMOD (Federal Republic of Germany). The report contains comprehensive descriptions of the various approaches and default values of most important parameters used in the models. Numerous remarks and comments are provided where the approach chosen may have weaknesses or faults or where the descriptions/manuals are incomplete, difficult to understand, or not consistent with information given in other documentation. An addendum is included with this report that provides SNL's assessment of all comments in the report related to MACCS 1.5. Since these comments relate directly to the suitability of the model implementation and documentation of MACCS 1.5, they are summarized here.

a. S. Acharya et al., *Benchmarking of the MACCS Code*, NUREG-1364, to be published.

The main problem encountered in the chronic exposure pathway models review was in regard to differences between the documentation and the model implementations. The review also noted numerous misprints and errors in the model descriptions in the draft of NUREG/CR-4184. SNL pointed out that the current version of NUREG/CR-4184 reflects the implementation of the intended chronic exposure pathway models as developed by Bob Ostmeier in 1983. Subsequent review by SNL of these models has led to changes in both the functional form of the models with many changes in parameter values in MACCS 1.5. NUREG/CR-4184 would need to be updated and corrected by SNL to correctly reflect the model implementation in MACCS 1.5. Other notations in the review regarding adequacy and completeness of the MACCS 1.5 documentation have been addressed in the revision of the User's Guide and the Model Description and completion of the Programmer's Reference Manual.

The review noted a few instances where changes made to models or to parameter values varied significantly from earlier implementations. This primarily involved the variables TCROOT and QROOT. SNL detailed the exact physical process represented by these two variables in MACCS 1.5 and explained what caused the change in implementation or recommended parameter values. There was some confusion in the implementation of the weathering equation in MACCS to which SNL provided a satisfactory explanation. Likewise, inhalation protection factors and groundshine weathering implementation were not as documented and required additional explanation by SNL to clarify them. Finally, deposition of radionuclides on water bodies is implemented in MACCS 1.5 in a manner different than the reviewer was familiar with. Again, SNL explained the implementation in sufficient detail to answer the reviewer's questions. Additional clarification regarding irreversible binding to soil and the soil ingestion transfer factor were provided by SNL to answer questions concerning these models.

The review does not point out any apparent errors in the MACCS 1.5 chronic exposure pathway models that could not be attributed directly to documentation errors. The primary conclusion of this review was that comprehensive documentation with contradictions removed did not exist for MACCS 1.5. With the revision to the Model Description and NUREG-4184, this would no longer be a problem.

5. CONCLUSIONS

The latest iteration of QA and verification of the MACCS 1.5 computer code shows that there are no obvious uncorrected errors in the code and that it is ready for release. The earlier reviews of the various code modules and documentation uncovered errors which could have affected MACCS 1.5 computational results. These have been fixed by the SNL code developers. Other items that have not as yet been addressed by SNL affect code readability, code maintainability and understanding of model implementation in MACCS 1.5. The primary items in this category include parameterization of common blocks and other program fragments, use of an INCLUDE facility to propagate changes through the code, a rewrite of the ATMOUT subroutine, a rewrite of the meteorological binning routines, and providing both the source codes and documentations for the dose conversion factors and

the maximum permissible ground concentrations preprocessor programs.

The draft documentation provided with MACCS 1.5 was reviewed and the later draft User's Guide was found to accurately represent the code. The draft Model Description and Programmer's Reference Manual contained numerous trivial errors that are expected to be corrected in the final documentation. Requested enhancements to the write-ups in the Model Description should help improve readability and understanding of the models implemented in MACCS 1.5. Additional documentation that currently exists only in draft form, if at all, relative to the MACCS input used for the NUREG-1150 revisions (NUREG/CR-4551) and the chronic exposure pathway models (NUREG/CR-4184) would provide valuable information for the preparation of MACCS 1.5 input and therefore should be completed and issued.

REFERENCES

1. U.S. NRC Office of Nuclear Regulatory Research, *Reactor Risk Reference Document*, NUREG-1150 (Draft), Volumes 1-3, February, 1987. (Available from Public Document Room, Washington, D.C.).
2. U.S. NRC Office of Nuclear Regulatory Research, *Severe Accident Risks: An Assessment for Five U.S. Nuclear Power Plants*, NUREG-1150 (Second Draft), Volumes 1 & 2, June, 1989.
3. *UNICOS User Commands Reference Manual*, SR-2011, Cray Research, Inc., Mendota Heights, MN, July 1987.
4. W. R. Cowell et al., *TOOLPACK/1 Release 2, Introductory Guide*, ANL-86-43, Argonne National Laboratory, October 1986.
5. *CFT77 Reference Manual*, SR-0018, Cray Research, Inc., Mendota Heights, MN, September 1986.
6. *FORTRAN 77 Analyzer User's Manual*, NBS GCR 81-359, TRW Defense and Space Systems Group, Redondo Beach, CA., 1981.
7. G. J. Myers, *Composite/Structured Design*, Van Nostrand Reinhold, New York, NY, 1978, p. 46.
8. M. H. Halstead, *Elements of Software Science*, Elsevier North-Holland, New York, NY, 1977.
9. T. J. McCabe, "A Complexity Measure," *IEEE Transactions on Software Engineering SE-2*, 6, December 1976, pp. 308-320.
10. D. B. Turner, *Workbook on Atmospheric Dispersion Estimates*, PB-191 482, U. S. Dept. of Health, Education and Welfare, Cincinnati, OH, 1970.
11. R. A. Dobbins, *Atmospheric Motion and Air Pollution*, John Wiley and Sons, New York, NY, 1972, p. 226.
12. U.S. Nuclear Regulatory Commission, *Reactor Safety Study Appendix VI: Calculation of Reactor Accident Consequences*, WASH-1400, NUREG 75/014, October 1975.
13. J. C. Helton et al., "Contamination of Surface-Water Bodies After Reactor Accidents by the Erosion of Atmospherically Deposited Radionuclides," *Health Physics*, 48, June 1985, pp. 757-771.
14. L. T. Ritchie et al., *CRAC2 Model Description*, NUREG/CR-2552, SAND82-0342, Sandia National Laboratories, 1984.
15. U. Tveten, *Review of the Chronic Exposure Pathway Models in MACCS and Several Other Well-Known Probabilistic Risk Assessment Models*, NUREG/CR-5377, Institutt for Energiteknikk, Postboks 40, N-20007 Kjeller, Norway, (Draft Available from Public Document Room, Washington, D.C.).