NUREG/CR-4369
SAND85-1774

# Quality Assurance (QA) Plan for Computer Software Supporting the U.S. Nuclear Regulatory Commission's High-Level Waste Management Program

Prepared by G. F. Wilkinson, G. E. Runkle

**Sandia National Laboratories**

**Prepared for**
**U.S. Nuclear Regulatory Commission**

NUREG/CR-4369
SAND85-1774

# Quality Assurance (QA) Plan for Computer Software Supporting the U.S. Nuclear Regulatory Commission's High-Level Waste Management Program

Manuscript Completed: December 1985
Date Published: January 1986

Prepared by
G. F. Wilkinson, G. E. Runkle*

Sandia National Laboratories
Albuquerque, NM 87185

*U.S. Department of Energy, Albuquerque, NM

**Prepared for**
**Division of Waste Management**
**Office of Nuclear Material Safety and Safeguards**
**U.S. Nuclear Regulatory Commission**
**Washington, D.C. 20555**
**NRC FIN A1166**

## ABSTRACT

A quality assurance plan has been developed for computer software created and/or maintained by Sandia National Laboratories, Division 6431, and subsequently transferred to the U.S. Nuclear Regulatory Commission in support of its high-level waste management program. The plan contains requirements for software storage and documentation, as well as a brief description of the program maintenance process. Division 6431 has established a Computer Software Management System for implementing this quality assurance plan.

CONTENTS

APPENDICES

## ACKNOWLEDGEMENTS

Several people made valuable suggestions and comments during the development of the QA plan and in the preparation of this document. In particular, the author wishes to acknowledge the suggestions of Robert Cranwell and Paul Davis of Sandia National Laboratories, and Walter Beyeler of OAO. Special thanks is given to Judy Mattingley in the typing of this document.

## 1.0 INTRODUCTION

The U.S. Nuclear Regulatory Commission (NRC) is assembling mathematical models for assessing demonstrations that the U.S. Department of Energy (DOE) is expected to make in order to show that its deep geologic repositories for high-level radioactive waste (HLW) are in compliance with NRC's HLW regulation, 10 CFR 60, and EPA's HLW Standard, 40 CFR 191. In support of this research, computer software has been developed for use in assessing the long-term risk from disposal of radioactive wastes in deep geologic formations, in estimating dose commitments and potential adverse health effects from released radionuclides, and in performing sensitivity and uncertainty analyses. Because of the potential use of this software in NRC licensing and regulatory activities, it is essential that a quality assurance (QA) program exist to ensure adequate quality in programs developed and in data generated by these programs, as well as for maintenance of the programs.

An ongoing computer software maintenance program requiring peer review and management approval should provide a systematic record of calculations and analyses that are performed, as well as a history of program changes. As an end result, availability of standard versions of computer software with complete documentation makes reproducibility feasible and associated research efforts defensible. This document describes the QA program for computer software created and/or maintained by Sandia National Laboratories (SNLA) Division 6431, and subsequently transferred to the NRC in support of the NRC's high-level waste management program. It was prepared in accordance with the "Draft Quality Assurance Plan for Operational Software" which was developed by R. Codell and S. Silling in the Division of Waste Management of the U.S. Nuclear Regulatory Commission and is intended to satisfy requirements outlined in NUREG-0856, "Final Technical Position on Documentation of Computer Codes for High-Level Waste Management." This QA program is primarily intended to address operational software (software which is in a usable state) and not the many varied stages of program development.

In addition to the guidelines established in this publication, the subject software must fulfill QA requirements within SNLA. All activities performed on projects within Organization 6000, the Energy Programs Vice Presidency, must comply with the Organization 6000 Quality Assurance Program Plan, which focuses on general policy and proposes uniform standards. The division supervisor and staff members have the formal responsibility for implementing the program. Each individual user must strictly adhere to the division guidelines in order for the QA program to be successful.

## 2.0 DEFINITIONS

2.1 Computer program - a procedure that has been coded for a computer and designed for solving a problem, including such problems as collection and processing of data and presentation of results. (Source: letter to E. Bonano from K. Goller, dated 9/26/84)

2.2 Documentation - a written description of a software package, including mathematical models, associated computer programs and specific calculations which has received peer review and management approval. (Source: letter to R. Cranwell from N. Coleman, dated 10/29/85)

2.3 Model - a representation of a process or system. (Source: letter to E. Bonano from K. Goller, dated 9/26/84)

2.4 Quality assurance of computer software - those planned and systematic actions necessary to provide adequate confidence that the software will serve its intended purpose. (Source: letter to R. Cranwell from N. Coleman, dated 10/29/85)

2.5 Retrievability - the capability to access all previous versions of software developed over a reasonable period of time even though they may not be executable due to hardware and/or system software changes.

2.6 Software - written or printed data, such as programs, routines, symbolic languages, and control languages essential to the operation of computers. (Source: letter to R. Cranwell from N. Coleman, dated 9/26/84)

2.7 Traceability - the ability to identify the actual configuration of the software used in the calculations. This characteristic is closely related to retrievability, which is described in detail above.

2.8 Validation - In waste management applications, the object of validation is the assurance that the model embodied in a computer program is a correct representation of the process or system for which it is intended. Validation is generally accomplished by using computer programs to simulate field or laboratory experiments. For cases where these experiments are well defined, validation is possible. However, parameters that control experiments involving groundwater flow and contaminant transport,

especially in fractured rock, at this stage are too poorly defined to allow for the validation of computer programs. In these cases, the exercise of applying the programs to field experiments is still a valuable test but falls short of "validation."

2.9 Verification - the process which demonstrates that the software correctly performs its stated capabilities. This is primarily done through a set of test problems designed to show that the stated equations are solved in a satisfactory manner, but not necessarily indicating that the model is a valid representation of any particular physical system. In addition, these problems can be used for familiarizing users with the function and execution of the computer program and providing a check between standard versions of the program on various computer systems.

## 3.0 COMPUTER SOFTWARE MANAGEMENT SYSTEM

A Computer Software Management System (CSMS) has been established in Division 6431 at SNLA to provide the mechanics for implementing QA for NRC-related computer software. The Division 6431 CSMS is administered by the QA Coordinator, a staff member designated by the Division Supervisor. This person has the necessary independence and authority to identify quality assurance problems, initiate or recommend corrective action, and verify the implementation of solutions. The QA Coordinator's authority extends to stopping work when nonconformances occur until corrective action is taken, subject only to review by the Division Supervisor. Decisions relative to quality-oriented matters are made at appropriate organizational levels, with the Vice President of Energy Programs having the ultimate responsibility for resolving all disputes.

The CSMS has the function to institute and maintain the following:

1. A repository for current and previously used versions of the computer software. This archiving system should maintain current versions of the computer software in an immediately accessible form, and previous versions in less accessible archived states. The archiving system must be sufficient to permit traceability and retrievability.

2. A repository for documentation about the software in the system, including original reports, documentation of errors, modifications and enhancements, results of verification test simulations, etc.

Any system command procedures and libraries of data required to utilize the programs should be present in the CSMS. The CSMS must also have methods for backing up all computer files and should maintain the documentation in a reasonably secure form for retrieval purposes. The QA Coordinator should make sure that the CSMS adapts to changing hardware and system software.

## 4.0 IMPLEMENTATION

This section describes the methods by which the CSMS will implement software storage and documentation.

### 4.1 Software Storage

#### 4.1.1 Archiving Programs for Retrievability

To ensure future retrievability, all computer software entering the Division 6431 CSMS will be placed into an archive file in a format that is compatible with operational computer equipment at that time. Each program must have a back-up file. Archive files cannot be destroyed without an audit to determine that retrievability requirements can still be satisfied.

Standard versions of programs are maintained on both the secure and open partitions of the SNLA computer Network Operating System (NOS). On a semi-weekly basis, everything on the SNLA computer system is copied to an active tape library. Consequently, at any given moment there exist two rotating back-up copies of a file. Archive tape files are created on a monthly basis. These archive tapes are stored in a climate controlled long-term facility. In addition, the QA Coordinator will establish in-house storage facilities for copies of each code.

#### 4.1.2 Program Listing

A program listing must be retained for each computer program version entered into the CSMS.

### 4.1.3 Verification Test Cases

The software containing a usable copy of verification test cases for each program version will be available in the CSMS.

### 4.1.4 Program Retrievability

The retrievability period for each inactive program shall remain indefinite to allow for a complete history of program modifications.

## 4.2 Documentation

Each computer program must satisfy reasonable requirements for documentation before use in calculations leading to published reports. Therefore, it is mandatory that the QA Coordinator be included in the manuscript review process for all reports generated by SNLA pertaining to the NRC's high-level waste program. In the QA Coordinator's absence, the division supervisor will have QA sign-off authority. (A copy of the manuscript review sheet is included as Appendix 1 to this document.) As previously mentioned in Section 3.0, it is essential that the QA Coordinator have the support of management to pursue resolution of a specific QA concern with higher management if necessary before release of the report.

It is the responsibility of the principal investigator to provide the program documentation. All scientific computer programs are to be documented in conformance with ANSI (American National Standards Institute) Standard N-413, "Guidelines for Documentation of Digital Computer Programs." In addition, large programs or complex systems should conform with FIPS Pub 38 (2/15/76), "Documentation of Computer Programs and Automated Systems." The guidelines provided in ANSI Standard N-413 have been expanded in order to suit more effectively the requirements of high-level waste licensing in NUREG-0856, "Final Technical Position on Documentation of Computer Codes for High-Level Waste Management." A brief description of documentation requirements is outlined below.

### 4.2.1 Internal Documentation

All software residing in the CSMS must contain internal documentation within the program as follows:

1) Program name and version identifier
2) Brief description of program
3) Original source of program

4) Name of author
5) History of modification including name of modifier, extent and date of modification
6) Proprietary details (if applicable)
7) Computer language and version
8) Necessary hardware
9) Documentation references

4.2.2 Traceability Documentation

All software residing in CSMS must produce its program name and version identifier on the program output to allow traceability.

4.2.3 Program Modifications

Diversified use of computer software typically results in the need to make changes to the original program, either through improvements or modifications. Improvements are made to correct coding or logic errors and invalid algorithms, as well as to refine the modeling approach or algorithm. Modifications to a program may be made when the user wishes to alter the modeling approach to accommodate a specific application.

Due to these changes, several versions of a program may be in existence. The standard version must be accompanied by a user's manual and self-teaching curriculum, and it must be compatible with the current computer operating system. The user's manual must satisfy requirements outlined in NUREG-0856, including a description of program considerations, data files, input data and system interface. Its primary purposes are to allow NRC staff to understand modeling results and to permit them to install and run the code on their own computer. In addition, verification (and validation if feasible) has generally been performed. The following programs developed for NRC research projects currently exist in standard version: SWIFT, DNET, NWFT, NWFT/DVM, LHSPL, DHEJL81, and PATH1.

For some projects and reports, it may be necessary to make modifications to a standard version program. A list of the modifications, the input data and the results, as well as the documentation required in Section 4.2.1, comprise a modified version. It is important to maintain a copy of the modified version so that, if necessary, the results may be regenerated at a later date for verification/validation purposes. If a new program is developed, the user's documentation, data and results must be maintained. Previous versions to the programs are archived and may not be compatible with the current computer operating system.

### 4.2.4 Formal Reports and Supporting Documentation

Documentation as described in Paragraph 2.2 must accompany all software residing in the CSMS. Supporting documentation will be required to define machine dependency, the current control statements, and any additional verification tests performed to satisfy Paragraph 2.9, and to record limitations or discrepancies (see Paragraph 4.2.5) that are not included in formal documentation. A complete explanation of the mathematical models and numerical methods used, as described in NUREG-0856, should also be included. The supporting documentation should show all efforts that contribute to software qualification, including discussion of improvements or corrections made to previous program version.

### 4.2.5 Reporting Program Discrepancies

Important elements in improving the quality of computer programs are identification and correction of observed errors, inaccuracies, and limitations in software and/or documentation. Reports of discrepancies covering program usage will be maintained by the program user and submitted to CSMS files. If significant discrepancies are reported, the program shall be re-evaluated for suitability. The principal investigator should take action to correct the problems and submit a new program version to the CSMS. The format for discrepancy reporting and disseminating the information to users is contained in Appendix 2 to this document.

### 4.2.6 Verification

Each program submitted to the CSMS must be verified as outlined in Paragraph 2.9 before use in calculations leading to published reports. Verification processes must include adequate tests to demonstrate that the program satisfactorily performs its stated capabilities. Since each program is different, an exact definition of the program testing requirements cannot be defined. Consequently, it is the reponsibility of the principal investigator to determine when a program has been adequately tested. The significant verification processes should be documented as test cases and stored in the CSMS as discussed in Paragraph 4.2.4.

### 4.2.7 Storage of Documentation

The QA Coordinator will establish facilities for storage of all required documentation, as well as tapes of all program versions. Pertinent information concerning each computer

program will be maintained in a designated QA notebook. Included for each program will be a summary description sheet which provides such details as program and documentation status, hardware and software requirements, and a brief summary of the purpose, scope and methods of the program. Additional information concerning data input requirements, output, program limitations or restrictions, program updates, details of methods used, etc. may be included if deemed relevant for satisfactory use of the program. (A completed copy of this form has been included as Appendix 3 to this document.) In addition, a program distribution form will be maintained to track versions of programs which have been transmitted to requestors. (A copy of this form is included as Appendix 4.) Records of modifications to existing programs will also be kept, as well as copies of the actual modifications. Appendix 5 is a copy of an update log to be maintained by the QA Coordinator. Persons making modifications to programs are to submit a description of their changes to the QA Coordinator on the Program Update Description Form, a copy of which is included as Appendix 6. Reference material for inactive programs will be maintained as long as the programs are maintained.

## 5.0 PROGRAM MAINTENANCE

### 5.1 Update Processor

It is imperative that records be kept of any alteration to a program, whether it be a permanent change to a standard program or a variation used for a specific application. Program maintenance is performed at SNLA through use of the Update processor on the Control Data Corporation (CDC) computer Network Operating System (NOS). It provides a history of program changes and is easily implemented on various CDC computer systems. Update, which is a utility for maintaining and editing a mass storage file containing images of coded punched cards or text lines, is also compatible with many computer operating systems (CDC, 1982), thus allowing ease of transfer of program changes. If a computer system is adopted by either the NRC or SNLA which does not support Update, alternative program maintenance means (such as the NOS/VE Source Code Utility) will be investigated.

A file of images to be manipulated by Update must be in a special format called a program library. Groups of line images within the program library are called decks. As each line image is written to the program library, Update assigns it a unique identifier (deck name) and sequence number which can be

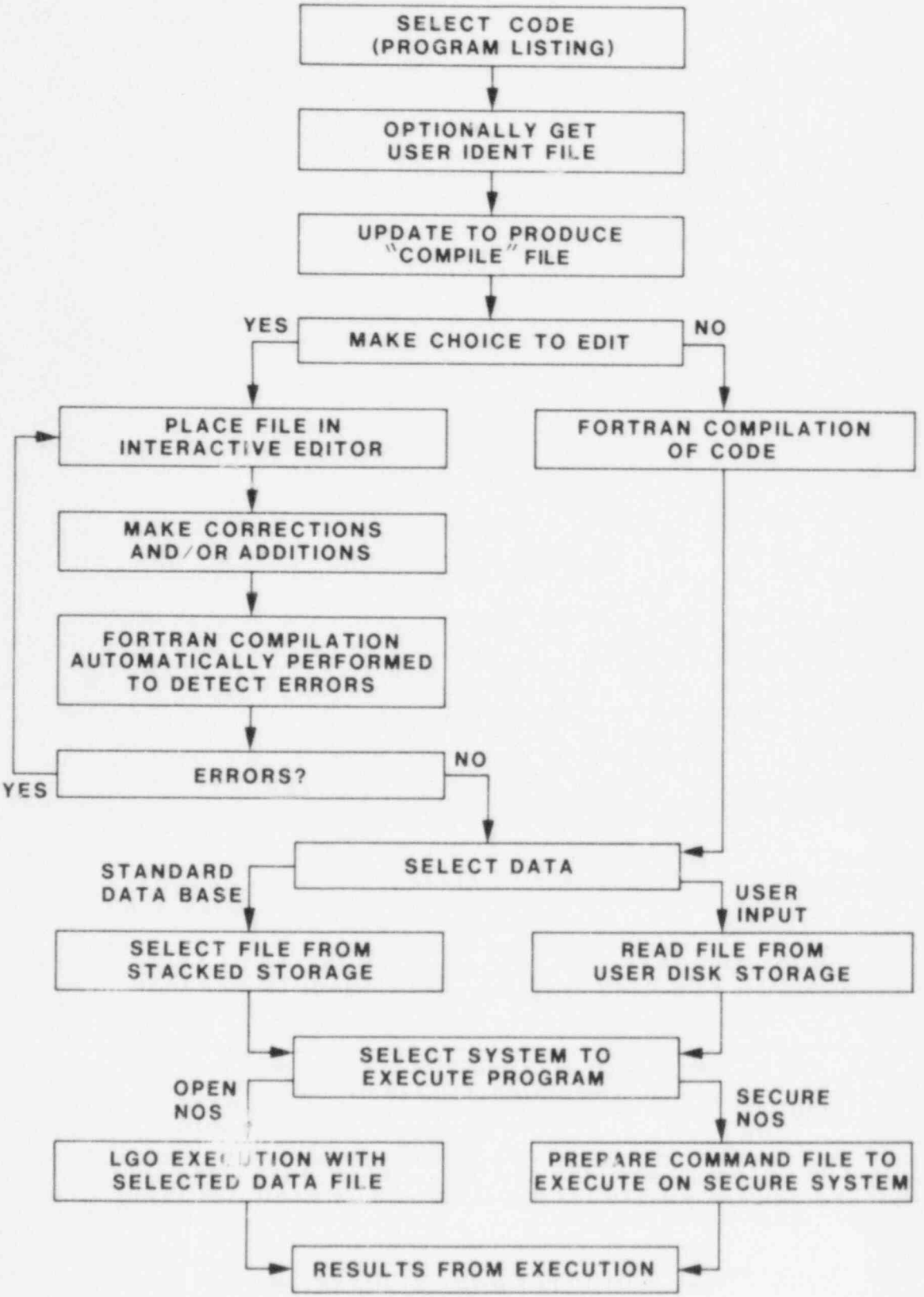


Fig. 1

INTERACTIVE SYSTEM

used to identify changes. A correction run manipulates the contents of the program library. These changes exist only for the duration of the run unless a new program library is created. Consequently, the original program library is never permanently altered.

A deck containing updates only can be extracted from the library and used as if it had been entered into the system as a punched deck. It is this deck, composed either of punched cards or images of punched cards, which is transferred to the NRC for updating their copy of the program. More specific information on the Update format is available in Update Version 1 Reference Manual, Publication No. 60449900, Rev. E, Control Data Corporation.

## 5.2 Interactive System

The Interactive System (IS) was developed to facilitate execution of standard programs that are maintained in the QUALIB library on the SNLA computer system. The IS, which operates on Open NOS, performs automatic Update compilation, provides an optional interactive edit mode, selects the prescribed data set, and executes the program on a designated computer system. Users with little knowledge of computer system commands can easily edit and execute programs. In addition, the system allows the QA Coordinator rapid access to the programs and permits execution of the standard data files to check and verify periodically results obtained from programs which have been modified.

The processes involved in the IS are outlined in the flow chart in Figure 1. Initially, the IS prompts the user for the name of the program to be used. This program is in the program library form, as was described in Section 5.1. The user may also optionally access an "IDENT" file containing Update-formatted modifications. The program library and optional "IDENT" file are then submitted to the Update processor, resulting in a "COMPILE" file.

The user may then choose to make modifications to the "COMPILE" file using an interactive editing routine. Following completion of editing, the file is submitted as the input file to the FORTRAN compiler to check for errors. If errors are found, the file is returned to the interactive editor for correction. If the user chooses not to edit, the program is sent directly to the FORTRAN compiler.

At this point, the user may select a standard data file or a file from the user's disk storage as input to the program. The standard data files consist of the sample problems from the user's manual for the specific program, from the self-teaching curriculum developed for the program, or from other documented analyses that used the respective program. The standard versions of data are stored in QUALIB in a "stacked" format, which makes optimal utilization of storage space. User-defined input must be in a format consistent with the standard versions as described in the supporting user's manual.

The user then chooses to execute the program on either the open or secure partition of the computing system. For computer programs that require large amounts of core, execution on the secure side is usually more efficient. However, the user must hold an active "Q" security clearance (issued by the DOE) in order to utilize the secure computer system. If the user selects the secure system, the IS creates a command file to be transferred and executed. The results from execution on either computer system are provided as output to a line printer, graphics, or other output device, as dictated by the program.

## 6.0 SOFTWARE DISTRIBUTION

"Requirements for Use of Microcomputers by NRC Contractors" and "Scientific Software Development, Distribution and Submittal Requirements for NRC Contractors" (Office of Resource Management, Division of Automated Information Services, U.S. Nuclear Regulatory Commission, 11/7/84) provide requisites for computer software developed by contractors for the NRC. The guidelines discussed in Section 4.0 were established commensurate with these requirements.

The QA Coordinator is responsible for determining that software has met these criteria and, consequently, no software should be submitted to the NRC prior to QA approval. Each software submittal to the NRC should be accompanied by the software summary form which is attached as Appendix 7 to this document. (This form is now being used in lieu of SF 185 which is required in NUREG-0856.)

As outlined in the above-mentioned documents, all computer programs developed with federal funds are to be made available for distribution to requestors through the National Energy Software Center (NESC) at Argonne National Laboratory. Consequently, standard versions of computer programs developed and/or maintained by SNLA for the NRC's HLW program will be made available for user dissemination through this source.

## REFERENCES


American National Standards Institute, "Guidelines for Documentation of Digital Computer Programs," ANSI Standard N-413 (American National Standards Institute).

Codell, R. and S. Silling, "Draft Quality Assurance Plan for Operational Software," (USNRC, 1984).

Coleman, N., USNRC, written communication (1985).

Control Data Corporation, "Update Version 1 Reference Manual," Publication No. 60449900, Rev. E. (1982).

Goller, K., USNRC, written communication (1984).

Henderson, J. T., Organization 6000 Quality Assurance Program Plan (QAPP) Rev. A, (Albuquerque, NM: Sandia National Laboratories, 1984).

Institute of Electrical and Electronics Engineers, Inc., "IEEE Standard for Software Quality Assurance Plans," IEEE Std. 730-1984, (New York, NY: IEEE, 1984).

Nunziato, J. W., M. E. Fewell, R. D. Krieg, D. W. Lobitz, B. J. Thorne, and M. P. Sears, Quality Assurance (QA) Policy for Computer Software in Directorate 1500, SAND83-0905 (Albuquerque, NM: Sandia National Laboratories, 1983).

Silling, S. A., Final Technical Position on Documentation of Computer Codes for High-Level Waste Management, NUREG-0856 (USNRC, 1983).

U.S. Department of Commerce/National Bureau of Standards, "Documentation of Computer Programs and Automated Systems," FIPS Pub 38 (USDOC, 1976).

U.S. Nuclear Regulatory Commission, "Requirements for Use of Microcomputers by NRC Contractors," USNRC, 1984).

U.S. Nuclear Regulatory Commission, "Scientific Software Development, Distribution and Submittal Requirements for NRC Contractors," (USNRC, 1984).

Appendix 1

Manuscript Review Sheet - 6400

Please review the attached manuscript:

Title: _____

Author(s) _____

Type of Publication:
( ) Formal (SAND_____)
( ) Conference to be presented at _____ _____ on _____
( ) Journal to be published in _____ _____

| | Org. | Initial[3] | Date |
|---|---|---|---|
| Reviewer:[1] _____ | _____ | _____ | _____ |
| Referee:[2] _____ | _____ | _____ | _____ |
| QA Coordinator (for 6431 use only) | _____ | _____ | _____ |

Please evaluate each on the basis of outstanding (1), very good (2), good (3), fair (4), poor (5).

| | Ratings | |
|---|---|---|
| | Reviewer | Referee |
| 1. Organization, clarity, and conciseness of the material presented. | _____ | _____ |
| 2. Correctness of any assumptions that are made. | _____ | _____ |
| 3. Validity of data including statistical significance. | _____ | _____ |
| 4. Adequacy of the discussion of variables. | _____ | _____ |
| 5. Validity of conclusions and recommendations. | _____ | _____ |
| 6. Adequacy of illustrations, graphs, tabular data, etc. | _____ | _____ |
| 7. Appropriate acknowledgment of contributors and referenced materials. | _____ | _____ |
| Overall Evaluation: | | |
| 1. How do you rate the paper? | _____ | _____ |
| 2. What course do you recommend? | | |
| a. Publish | _____ | _____ |
| b. Publish following minor revisions | _____ | _____ |
| c. Publish only after major revision | _____ | _____ |
| d. Reject | _____ | _____ |
| 3. Is there anyone else who should see the manuscript before it is released for publication? | _____ | _____ |
| 4. Comments (see reverse of this form). | | |



(over)

Please make comments, initial your comments, and return this form to the author (s).

Reviewer Comments:

Referee Comments:

Author: This form must be included as part of the Review and Approval package.

Appendix 2

SOFTWARE PROBLEM REPORT FORM

This form should be filled out for every software problem encountered by users. Please provide a copy to the program's Task Leader or Principal Investigator.

Date_____

Name and Organization of Preparer:_____

Program Name:_____Program Version Designator_____

Reason for Report:

Details: (Attach items as needed to support reason for report)

Distribution (To Affected Users):

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Appendix 3

SA 7198-A(3-82)

## SUMMARY DESCRIPTION OF COMPUTER CODES

| | | |
|---|---|---|
| 01. SUMMARY ISSUE DATE: YR MO DAY: 83 - 07 - 15 | 02. SOFTWARE TITLE: The Network-Flow and Transport Model | 03. CODE START DATE: YR MO DAY: 78 - _ - _ |
| 04. SUMMARY REVISED DATE: YR MO DAY: _ - _ - _ | | 05. CODE FINISH DATE: YR MO DAY: 79 - _ - _ |
| 06. SUMMARY PREPARED BY: (NAME/PHONE NO.) R. M. Cranwell 844-8368 | 07. SHORT TITLE (IDENTIFICATION): NWFT | 08. CODE REVISION DATE: YR MO DAY: _ - _ - _ |
| | | 09. NRC IDENTIFICATION NUMBER: |

| | |
|---|---|
| 10. AUTHORS: Campbell, J. E., Kaestner, P. C., Langkopf, B. S., and Lantz, R. B. | 11. SANDIA CONTACT (ADDRESS & PHONE): R. M. Cranwell, Division 6431, 844-8368 |

12. KEY WORDS: Network model, groundwater transport, contaminant transport

| 13. COMPATIBLE COMPUTER SYSTEM(S): | 14. PROGRAMMING LANGUAGE(S): | 15. OPERATING SYSTEM REQUIREMENTS: | 16. NUMBER OF SOURCE PROGRAM STATEMENTS: |
|---|---|---|---|
| CDC, CYBER, CRAY, VAX | FORTRAN (100%) | SCOPE 2.1 | |

| 17. CODE STATUS: | 18. USER'S MANUAL STATUS: | 19. SELF-TEACHING CURRICULUM STATUS: | 20. SOFTWARE AVAILABILITY: |
|---|---|---|---|
| ☐ PROPOSED | ☐ NOT INTENDED | ☐ NOT INTENDED | ☐ AT ARGONNE CODE LIBRARY |
| ☐ IN PROGRESS | ☐ PROPOSED | ☐ PROPOSED | ☒ AVAILABLE |
| ☒ COMPLETED | ☐ IN PROGRESS | ☐ IN PROGRESS | ☐ LIMITED |
| | ☒ COMPLETED* | ☒ COMPLETED* | ☐ IN-HOUSE ONLY |
| *SEE REFERENCES | | | ☐ PROPRIETARY |

SUMMARY DESCRIPTION OF COMPUTER CODES Side 2 (Continued from side one)

21. MACHINE REQUIREMENTS

Core is minimal.

22. SUPPORT MATERIAL

Requires a linear systems solver. Current coding calls for the Gaussian elimination routine LEQTIF of the IMSL package.

23. PROGRAM/CODE SUMMARY (INCLUDE PURPOSE, SCOPE, AND GENERAL METHOD) *SHOULD NOT EXCEED 500 WORDS*

PURPOSE: The code is a simulator for groundwater flow and contaminant transport. Due to its simplicity and extreme efficiency, it is amenable to multiple runs and is thus recommendable for statistical studies such as sensitivity analyses.
SCOPE: For groundwater flow, a network of one-dimensional segments with constant pressure boundary conditions is used. User must decide if the network is sufficient to represent the flow field of interest. The radionuclide transport model treats chains of up to length three with equal retardations. The source strength is a decaying band and transport is assumed through porous media.
GENERAL METHOD: The flow calculations are accomplished by using Darcy's law for each individual segment coupled with conservation of flow at segment junctions. A linear system of equations in the unknown pressures is solved. The result is used to determine head changes across a segment, hence flow and velocity. The transport model uses an average velocity from source to discharge. The solution to the convective-dispersion equation in the dimension is accomplished using sums of complementary error functions.

24. TYPICAL RUN TIMES

Run time is less than one second for any problem.

25. REFERENCES:

Campbell, J. E., et al., Risk Methodology for Geologic Disposal of Radioactive Waste: The Network Flow and Transport (NWFT) Model, SAND79-1920, NUREG/CR-1190, 1980.

Finley, N. C., Campbell, J. E., and Longsine, D. E., NWFT Self-Teaching Curriculum, SAND81-0372, available and NRC/PDR.

26. APPENDICES (SHOULD BE TYPED IN THE FOLLOWING ORDER ON SUPPLEMENTAL SHEETS) *Check Box if Appendix included in Code Summary.*

- ☒ A. DATA INPUT REQUIREMENTS
- ☒ B. OUTPUT
- ☒ C. CODE RESTRICTIONS
- ☐ D. CODE UPDATES
- ☐ E. DETAILS OF METHOD(S) USED INCLUDING EQUATIONS, NOTATIONS, ETC.
- ☐ F. **
- ☐ G. **
- ☐ H. **

** *PREPARER OF SUMMARY DESCRIPTION MAY INCLUDE OTHER APPENDICES AS APPROPRIATE.*

## Appendix A

## Data Input Requirements

Input categories

By leg: conductivity (ft/day), cross-sectional area ($ft^2$), length (ft), porosity (dimensionless),

By junction: pressure head at the inlets and outlet (ft), elevation (ft),

By isotope: half-life (y), initial inventory (ci),

By isotope and leg: distribution coefficients ($ft^3/lb$),

Miscellaneous: leach time (y), dispersivity (ft), problem time (y).

## Appendix B

## Output

Output from the network consists of a volumetric flow rate, a specific discharge, and an interstitial velocity for each leg. Output from the transport model can be in terms of time-dependent discharge rates, peak discharge rate, or integrated discharge.

## Appendix C

## Code Restrictions

The number of legs and junctions of the flow network is fixed. So, the flow field proposed by the user must be amenable to the current network. Only saturated porous flow is considered. The number of isotopes in a decay chain must be less than or equal to three and all must have the same retardation factor.

Appendix 4

SOFTWARE DISTRIBUTION

| Date | Requestor | Program | Version | Tape ID | Listing Location | Comments |
|---|---|---|---|---|---|---|
| | | | | | | |

Appendix 5

UPDATE LOG

| Date | Program | Version | Update No. | File Name | Date Sent to NRC | Description of Update |
|---|---|---|---|---|---|---|
| | | | | | | |

Appendix 6

SOFTWARE UPDATE FORM

Date:

Computer Program and Version:

File Name:

Update No.:

Person responsible for Update:

Explanation of Update:

Appendix 7

NRC Scientific Software Submittal Package Description Form

Program Name:

Programming Language(s) Used:

Machine:

Core Storage Requirements for Sample Problem:

Approximate Execution Time for Sample Problem:

Package Contents (items 1-8 are mandatory):

_____1. Abstract

_____2. Documentation consisting of the items described in the space below:

_____3. Source program on ____ cards ____ tape ____diskette.

_____4. Sample problem input on____ cards ____ tape ____ diskette.

_____5. Compilation of the source program (item 3).

_____6. Listing of the sample problem input (item 4).

_____7. Output from an execution of the sample problem input, item 4, using the source code provided in item 3, including plots, if any.

_____8. For tape submittals, a copy of the output for the job that created the tape.

_____9. Other (describe other materials such as data libraries, control information, etc., in the space below).

Distribution:

U.S. Government Printing Office
Receiving Branch (Attn: NRC Stock)
8610 Cherry Lane
Laurce, MD 20707

3141 C. M. Ostrander (5)
3151 W. L. Garner
6400 A. W. Snyder
6430 N. R. Ortiz
6431 R. M. Cranwell
G. F. Wilkinson (30)
8024 M. A. Pound

NRC FORM 335 (2-84) NRCM 1102, 3201, 3202

U.S. NUCLEAR REGULATORY COMMISSION

# BIBLIOGRAPHIC DATA SHEET

SEE INSTRUCTIONS ON THE REVERSE

1. REPORT NUMBER (Assigned by TIDC, add Vol. No., if any)

NUREG/CR-4369
SAND85-1774

2. TITLE AND SUBTITLE

Quality Assurance (QA) Plan for Computer Software Supporting the U.S. Nuclear Regulatory Commission's High-Level Waste Program

3. LEAVE BLANK

4. DATE REPORT COMPLETED

MONTH: December YEAR: 1985

5. AUTHOR(S)

G.F. Wilkinson, G.E. Runkle

6. DATE REPORT ISSUED

MONTH: January YEAR: 1986

7. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Sandia National Laboratories
Albuquerque, NM 87185

8. PROJECT/TASK/WORK UNIT NUMBER

9. FIN OR GRANT NUMBER

FIN A-1166

10. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS (Include Zip Code)

Division of Waste Management
Office of Nuclear Material Safety and Safeguards
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555

11a. TYPE OF REPORT

Final

b. PERIOD COVERED (Inclusive dates)

12. SUPPLEMENTARY NOTES

13. ABSTRACT (200 words or less)

A quality assurance plan has been developed for computer software created and/or maintained by Sandia National Laboratories, Division 6431, and subsequently transferred to the U.S. Nuclear Regulatory Commission in support of its high-level waste program. The plan contains requirements for software storage and documentation, as well as a brief description of the program maintenance process. Division 6431 has established a Computer Maintenance System for implementing this quality assurance plan.

14. DOCUMENT ANALYSIS - a. KEYWORDS/DESCRIPTORS

software quality assurance
software documentation

b. IDENTIFIERS/OPEN-ENDED TERMS

15. AVAILABILITY STATEMENT

Unlimited

16. SECURITY CLASSIFICATION

(This page) Unclassified

(This report) Unclassified

17. NUMBER OF PAGES

18. PRICE

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







NUREG/CR-4369

QUALITY ASSURANCE (QA) PLAN FOR COMPUTER SOFTWARE SUPPORTING THE U.S. NUCLEAR REGULATORY COMMISSION'S HIGH-LEVEL WASTE MANAGEMENT PROGRAM

JANUARY 1986