

WM-1  
PDR

Return to WM  
623-55

JUL 08 1982

WMHL: 3108.5

MEMORANDUM FOR: Malcolm R. Knapp  
High-Level Waste Licensing  
Management Branch  
Division of Waste Management

FROM: Stewart A. Silling  
High-Level Waste Licensing  
Management Branch  
Division of Waste Management

SUBJECT: CHANGES TO TECHNICAL POSITION ON DOCUMENTATION OF MODELS

The attached document is a revised version of the technical position on documentation of models, reflecting all comments received on the draft. I intend that this revised version will become the final technical position, subject to NRC management review.

ORIGINAL SIGNED BY

Stewart A. Silling  
High-Level Waste Licensing  
Management Branch  
Division of Waste Management

Enclosure:  
as stated

Distribution: w/o encl. WM-82-382

- ✓ WMHL file
- WMHL r/f
- WM r/f
- NMSS r/f
- JBMartin
- REBrowning
- MJBell
- MRKnapp
- SASilling & r/f

JOBunting  
HJMiller  
PDR

8208040013 820708  
PDR WASTE  
WM-1 PDR

OFFICE ▶	WMHL <i>SAS</i>					
SURNAME ▶	SASilling					
DATE ▶	7/5/82					

DOCUMENTATION OF COMPUTER CODES  
FOR HIGH-LEVEL WASTE MANAGEMENT

I. INTRODUCTION

The NRC anticipates that DOE will make use of computer codes in providing the "reasonable assurances" required by 10 CFR 60. Accordingly, it is necessary for NRC to evaluate these codes in its licensing review. The purpose of NRC's evaluation is to ensure that the codes are based on sound physical and mathematical principles, and that these principles are correctly applied.

In order to provide the NRC staff with an adequate basis for this evaluation, complete documentation of the codes is necessary. This technical position describes guidelines for documentation of the codes used by the applicant in performing the analyses submitted in support of a license application under 10 CFR 60.

As a basis, the guidelines draw on ANSI (American National Standards Institute) Standard N-413, Guidelines for the Documentation of Digital Computer Programs. However, many items in the ANSI standard are expanded upon in order to better suit the requirements of high-level waste licensing. In particular, this technical position calls for theoretical documentation in much more depth than the ANSI standard requires. This more detailed treatment of the basic methods is necessary for high-level waste licensing applications because of the anticipated importance that modeling will have in licensing decisions and because certain codes that will be used are relatively new.

The documentation called for is divided into five categories:

- (1) Software Summary
- (2) Description of mathematical models and numerical methods
- (3) User's manual
- (4) Code assessment and support
- (5) Continuing documentation and code listings

The software summary is a one-page document identifying the code and its purpose; a form is provided in Exhibit 1.

The documentation on mathematical models and numerical methods will provide the basis for NRC's review of the theory and means of solution used in the code. It should contain derivations and justification for the model.

Mathematical models and numerical methods have been combined into one category because in many modeling procedures the two aspects are so closely related that documenting them separately is inconvenient. However, separate documentation of the mathematical model and its numerical representation is acceptable, providing the documents are adequately cross-referenced.

The user's manual has two functions for licensing review. First, it helps the NRC staff in understanding modeling results which are submitted by the applicant during the licensing process. Second, it permits NRC to install and use the code on its own computer.

The documentation on code assessment and support describes to what extent and how the code has been validated and verified. It also documents the various code versions in use and the steps being taken to control and maintain these versions. Its purpose in licensing is to keep all parties aware of the practical limitations of the code, to record application-oriented modifications, and to help ensure that the code is as error-free as possible.

The final category, continuing documentation, provides a means for informing NRC of changes in the codes and what is known about them. It includes a provision for error reporting. It also includes revisions of the first four categories of documentation. This category includes computer-readable and paper source listings of the current version and new versions as they are released.

The guidelines stated in this technical position refer primarily to content rather than format. It is not necessary to write new documents if documentation exists which contains the information called for. It is acceptable to augment existing documentation which is missing some of the information required.

The guidelines below are designed for scientific, engineering, and mathematical codes. In developing the guidelines, it was assumed that the codes are batch-oriented rather than interactive. If interactive codes are to be used, appropriate changes should be made in the documentation.

## II. DEFINITIONS

Model - A representation of a process or system.

Mathematical model - A mathematical representation of a process or system.

Component model - A logically distinct subset of a model.

Numerical method - A procedure for solving a problem primarily by a sequence of arithmetic operations.

Numerical model - A representation of a process or system using numerical methods.

Computer code - A set of computer instructions for performing the operations specified in a numerical model.

Verification - Assurance that a computer code correctly performs the operations specified in a numerical model.

Validation - Assurance that a model as embodied in a computer code is a correct representation of the process or system for which it is intended.

## III. DOCUMENTATION GUIDELINES

### A. Software Summary

The software summary is a one-page document which identifies the code and version number and gives other basic information.

A new software summary should be sent to NRC when a new version is released or when major modifications are made. NRC should be notified by letter or through submittal of a new software summary when the people identified change. (See part E).

A form for the software summary is provided in Exhibit 1. The form may also be obtained in Federal Information Processing Standard Publication 30 (FIPS PUB 30), available for sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. Forms are also available as a GSA Federal Supply Stock Item, FSN 7540-118-8541.

## B. Mathematical Models and Numerical Methods

This section should provide a complete explanation of the methods used. As described below, it should include a derivation and justification for the model and should clearly explain its capabilities and limitations. It should contain extensive references to publications and point out places where new procedures were developed for the code being documented. This section should be complete enough to stand alone as a basis for review of the methods used in the code.

The discussions of mathematical models and of numerical methods may be separated into different sections provided they are adequately cross-referenced.

(1) Statement and description of the problem. Describe the overall nature and purpose of the general analysis in which the model will be used. State which specific aspects of this analysis the model will be used for. Indicate in general terms the information which goes into and comes out of the model. Show how the results of the model will be used in the overall analysis.

(2) Structure of the system model. If the complete model is comprised of smaller component models, as is frequently the case in large codes, describe briefly the role of each component model. Show how each component model contributes to the overall solution of the problem. Use flowcharts and block diagrams to describe the mathematical solution strategy.

(3) General numerical procedure. Describe the numerical solution strategy and computational sequence. Use flowcharts and block diagrams. Give references for the basic numerical procedure. If the method solves a large set of linear equations, show the structure of the equations and how the coefficients are determined. Show how the numerical strategy is related to the mathematical strategy (e.g., how boundary conditions are introduced).

(4) Component models. For each component model, provide the following:

(a) Purpose. Describe the purpose and scope of the component model. State in general terms the input to and output from the model and the way the information is processed. If the model is used only in certain cases, state under what circumstances it is used.

(b) Assumptions and limitations. Describe the assumptions and limitations of the component model, including simplifying assumptions about the geometry and behavior of the system. Include the known ranges of validity of the model for all variables. For empirical or semi-empirical models, state the range and type of data on which the model is based (e.g., field or laboratory data). State any known uncertainty about the validity of the model.

(c) Notation. Identify all algebraic variables used in equations. Give the mathematical symbols used in both the fundamental equations and their equivalents in the numerical formulation. Also give the computer variable name associated with each quantity.

(d) Derivation. Give a reference for the original publication in which the component model appeared. Give any additional references in which modifications of the model were presented that led to the final form. Start the derivation from generally accepted principles. Justify each step in the derivation and note how assumptions and limitations are introduced. For empirical and semi-empirical models, describe how experimental data was used to arrive at the final form. Clearly state the final mathematical form of the model. It is acceptable to provide references to published material containing a derivation in lieu of a new derivation.

(e) Application. Discuss the applicability of the component model to a geologic repository. Point out any extrapolation of the model or use out of range. Describe any restrictions on the use of the model, e.g., to a particular rock type. State whether the validity of the model will be affected by unusual or extreme conditions, such as high repository temperatures.

(f) Numerical method type. Characterize any numerical method used in the model which goes beyond simple algebra (e.g., finite-difference, Simpson's rule, cubic splines).

(g) Derivation of numerical model. Derive the numerical procedure from the mathematical component model. Give references for all numerical methods. Give the final form of the numerical model and explain the algorithm. If the numerical model produces only an intermediate result, such as terms in a large set of linear equations which is later solved, explain how the intermediate results are used. State what variables are input to and output from the component model.

(h) Location. State where the component model is located within the code. Refer to the code listing which will be part of the documentation. (See part E(6)).

(i) Numerical stability and accuracy. Discuss the stability and accuracy of the numerical model. Distinguish between those aspects of stability and accuracy which have been proven mathematically and those which have not been proven but have been observed in practice.

(j) Alternatives. Briefly discuss alternatives to the component model and state why this one was selected.

(5) Experience. Discuss the overall performance of the entire model. Note the conditions under which the model gives good or bad results. Point out specific component models known to perform poorly under certain circumstances. Give any general rules or recommendations to follow in use of the models.

### C. User's Manual

The user's manual, for purposes of licensing review, plays two roles. First, it allows NRC staff to understand modeling results submitted by the applicant. Second, it permits NRC to install and run the code on its own computer. The user's manual together with the code listing required in part E should be sufficient to instruct the user on how to set up and run problems as well as resolve possible difficulties.

If any of the items below are documented fully in comment cards in the code or by self-documenting features, the lines in the code which contain these comment cards or features may simply be referred to in place of a detailed description in the user's manual. However, the information must be complete enough to enable a new user with a reasonable amount of programming experience to run the code and understand errors.

#### (1) Program considerations.

(a) Program options. Discuss the function of each major program option; give special attention to effects of combinations of options. Relate options to the input values which control them.



(b) Program paths. Describe the purpose of each subroutine. Use flowcharts and block diagrams to explain the paths the program can take. If parts of the code are executed only under certain conditions, state those conditions. Show how the computational sequence and solution strategy described in part B(3) are related to the program flow.

(c) Data structures. Discuss how data are stored during computation. Describe the purpose and content of important common blocks and arrays. State the array dimensions. If dynamic dimensioning is used, describe the indexing algorithm. This section, together with a code listing, should be sufficient to allow the user to follow the flow of important data through the computational sequence.

(d) Initialization. Provide any values automatically assigned to important variables. These should include both values of physical significance and parameters which only affect program execution. State where the values are initialized and whether they are default or fixed values.

(e) Restart procedures. Describe any restart capabilities of the code and how they are used.

(f) Error processing. Describe the points of origin and likely causes of all major error messages, error switches, and abnormal stops.

(2) Data files.

(a) Content. Outline the general content, purpose, and organization of each data file.

(b) Use by program. Describe how and when the files are read and written by the program.

(c) Auxiliary processing. Describe any available auxiliary programs which create, modify, or use the files.

(3) Input data.

(a) General considerations.

(i) Techniques. Describe special input techniques and requirements; e.g., blank field treatment, order of items, field delineation.



(ii) Consecutive cases. If the code is able to retain input data from previous cases, give conditions for retention and reinitialization.

(iii) Defaults. Give the general conventions governing default values.

(b) Individual input records.

(i) Record identifier. Give the line identifier, if any, for this type of record.

(ii) Input variables. State the code variables which will contain data given on this record.

(iii) Format. Specify the format of this record, if any.

(iv) Need. For each variable, specify whether input is necessary or optional for both start and restart runs.

(v) Repetition. State how many of these input records are required or may be used optionally.

(vi) Units. For each input field, state the dimensional units.

(vii) Default. State the default value, if any, for each field.

(viii) Description. Define the meaning of each variable and discuss its primary use within the code. State how the user should assign values in setting up a run.

(ix) Range. State the acceptable limits for each variable necessary for successful operation of the model in the programming sense.

(4) System interface.

(a) System-dependent features. List the external references in the program which must be supplied by the system, and state the purpose of each. Include plot and mathematical libraries, but omit standard Fortran intrinsic functions.

(b) Compiler requirements. Specify what compilers have been used and any special load options that are necessary. Include compiler options, such as indirect large core memory addressing.

(c) Hardware requirements. Describe any special features needed. State the amount of central memory required for a typical case, or give an algorithm for determining the required amount of memory.

(d) Control cards or command files. All computer codes require some system commands to control program initiation, manipulation of files, and interaction with other programs. Describe the control cards or command files necessary to run the code as part of a modeling analysis. The appropriate level of detail will depend on the degree to which control cards or command files contain logic affecting program flow, manipulation of files, and communication among programs. Give sample control card files or command files. Discuss application dependence.

(5) Output. Discuss the code output. Relate edited output to input options, and state the origin and meaning of the output variables. Describe any normalizations of results and list associated dimensional units. Describe any graphical capabilities of the code.

(6) Sample problems. Choose a few problems which demonstrate how the code is used. These problems need not have known solutions or experimental data. However, they should exercise a large portion of the available programmed options and should use only a reasonable amount of computer time. Input deck listings and sample output should be given.

D. Code Assessment and Support

This document has the purpose of describing all work which sheds light on the adequacy of the code. The goal of the document for licensing purposes is to ensure that the code has been extensively reviewed and validated. In addition, the document describes steps the applicant is taking to ensure that code performance will not be degraded by future changes.

(1) Model review. Describe any projects for independent review of the methods used in the model. Include past and ongoing programs as well as planned ones. Summarize the results of past reviews. Describe any plans for modification of the model as a result of the reviews. Give references for publications.

(2) Verification and validation. Describe the program for verification and validation of the code. Include past, ongoing, and planned future activities. Give descriptions, input files, and results for specific tests. The input decks should be appropriate for the code version delivered to NRC (see part E). State what aspects of the code each test demonstrates and discuss how well the code performed. Describe any plans for modification of the model as a result of the tests. It is acceptable to provide reproductions of publications, output, and reports documenting the tests. The above information should be given for two classes of tests:

- (a) tests by the code developer; and
- (b) independent assessment.

(3) Maintenance and quality assurance. Often there are several versions of a code, each with different modifications, capabilities, and limitations. For licensing purposes, it is not necessary to completely verify and validate each version as though it were a separate code. However, it is necessary to ensure that each version is as correct as possible and that an orderly procedure exists for keeping track of the differences between versions. Since this procedure relates directly to the adequacy of the code for licensing purposes, provide a description of the maintenance and quality assurance programs for the code. Include a brief chronology of the code versions.

E. Continuing Documentation and Code Listings.

Codes generally continue to evolve even after a standard version has been released. This evolution includes the development of new capabilities, the detection and repair of errors, and application-oriented modifications. In order to ensure that licensing decisions are made on the basis of up-to-date information, it is necessary that NRC be kept informed of changes in the codes.

Continuing documentation includes revision of the other documentation and error reporting. It also includes computer-readable and paper listings of the current version and new versions as they are released.

(1) Updated software summaries. Submit a new software summary when the information it contains changes, when new code versions are released, or when important changes in the code are made.

(2) Technical contact. Identify a person whom the NRC staff can contact directly with technical questions about installing and running the code. NRC should be informed whenever a new person is given this responsibility.

(3) Documentation revisions. Revise the documentation sent to NRC as needed when changes are made in the code, when errors are found in the existing documentation, when new limitations of the model are found, or when new results of the assessment program described in part D appear.

(4) Error reporting. When errors or omissions that could affect the validity or appropriateness of model itself or specific instances of its use are found, these errors must be reported to NRC promptly. This includes input errors. Report action taken to correct the errors. State the significance of the error in past, current, and future modeling activities.

(5) Computer files. Send NRC computer files containing the current code version, new versions as they are released, and necessary updates as they are determined. Include the input decks for the sample problems (see part C (6)). Include all necessary library routines. The means of transmittal may be any reasonably standard medium, although 9-track magnetic tapes are preferred. The information should be in a standard format readable by a variety of computer systems. All files should be accompanied by a printout from the runs which created them.

(6) Paper listings. Provide printed listings of the current version, new versions as they are released, and updates. Code listings should include line numbers from UPDATE or another text processing utility.

(7) Description of updates and new versions. All updates and new versions delivered to NRC should be accompanied by descriptions of the changes.

(8) Response to NRC questions. Provide responses to questions by NRC concerning the model, its use, or installation. Questions will state whether written response, response by telephone, or a meeting is required.