ASTM

Designation: E 1472 – 92 (Reapproved 1998)

An American National Standard

Standard Guide for
Documenting Computer Software for Fire Models[1]

This standard is issued under the fixed designation E 1472; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ϵ) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This guide provides information that should be in documentation for computer software prepared for scientific and engineering computations in fire models and other areas of fire protection engineering.

1.2 The guidelines are presented in terms of three types of documentation: *(1)* technical document; *(2)* user's manual; and *(3)* installation, maintenance, and programming manual.

1.3 *This standard does not purport to address all of the safety concerns, if any, associated with its use. It is the responsibility of the user of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.*

2. Referenced Documents

2.1 *ASTM Standards:*

E 176 Terminology Relating to Fire Standards[2]

E 919 Specification for Software Documentation for a Computerized System[3]

E 1013 Terminology Relating to Computerized Systems[3]

E 1355 Guide for Evaluating the Predictive Capability of Fire Models[2]

2.2 *ANSI Standards:*

ANSI X3/TR-1 American National Dictionary for Information Processing[4]

ANSI/X3.88 American National Standard for Computer Program Abstracts[4]

ANSI/ANS 10.2 Recommended Programming Practices to Facilitate the Portability of Scientific and Engineering Computer Programs[4]

ANSI/ANS 10.3 Guidelines for the Documentation of Digital Computer Programs[4]

ANSI/ANS 10.5 Guidelines for Considering User Needs in Computer Program Development[4]

ANSI/IEEE 729 IEEE Standard Glossary of Software Engineering Terminology[4]

ANSI/IEEE 1063 Software User Documentation[4]

3. Terminology

3.1 *Definitions*—Definitions used in this guide are in accordance with Terminologies E 176 and E 1013, unless otherwise indicated. ANSI X3/TR-1 and ANSI/IEEE 729 include definitions of some technical terms used in this guide.

4. Significance and Use

4.1 This guide provides recommendations for writers of user's manuals and other documents for computer software prepared for scientific and engineering computations in fire models and other areas of fire protection engineering. The guide provides information that can be included in terms of three types of documents.

4.2 This guide is intended to assist in the understanding, usage, transfer, conversion, and modification of computer software. If the options and instructions contained in this guide are considered when documentation is prepared, the software should be used more readily for its intended purposes.

4.3 The use of fire models currently extends beyond the fire research laboratory and into the engineering, fire service, and legal communities. Sufficient documentation of computer software for fire models is necessary to ensure that users can judge the adequacy of the scientific and technical basis for the models, select the appropriate computer operating environment, and use the software effectively within the specified limitations. Adequate documentation will help prevent the unintentional misuse of fire models.

4.4 Specification E 919 is a standard specification for the documentation of computer software. It can be used to specify the minimum computer program documentation that shall be provided by the supplier of software to the purchaser.

4.5 Additional guidelines on documentation can be found in ANSI/ANS 10.3 and ANSI/IEEE 1063.

4.6 ANSI/ANS 10.2 and 10.5 provide guidelines for programming to ease the portability of the software and meet user needs.

5. Types of Documents

5.1 *General:*

5.1.1 There are many levels of desirable documentation, ranging from that needed by the user who wants only to run the programs, to documentation needed by the user who intends to make extensive modifications or additions to the programs.

5.1.2 This guide provides suggestions for items to include in

---

[1] This guide is under the jurisdiction of ASTM Committee E-5 on Fire Standards and is the direct responsibility of Subcommittee E05.33 on Fire Safety Engineering.
Current edition approved May 20, 1992. Published June 1992.

[2] *Annual Book of ASTM Standards,* Vol 04.07.

[3] *Annual Book of ASTM Standards,* Vol 14.01.

[4] Available from American National Standards Institute, 11 West 42nd Street, 13th Floor, New York, NY 10036.

three types of documents: *(1)* technical document; *(2)* user's manual; and *(3)* installation, maintenance, and programming manual. The items suggested for these manuals can be combined into a single document.

5.1.3 The documents should be written and organized to reflect the expected sophistication of the user.

5.2 *Technical Document*—This type of document is intended for use by the individual interested in an in-depth explanation of the scientific basis for the model. Articles in scientific or engineering journals are examples of this type of document.

5.3 *User's Manual*—This self-contained manual is directed to the prospective user of the fire model. With this type of manual, the user of the model should be able to understand the model application and methodology, reproduce the computer operating environment and the results of sample problems included in the manual, modify data inputs, and run the program for specified ranges of parameters and extreme cases. The manual should be concise enough to serve as a reference document for the preparation of input data and the interpretation of results.

5.4 *Installation, Maintenance, and Programming Manual*—This type manual is for the individual responsible for implementing the program on a computer, modifying or extending it to meet local needs, converting it to a different computer environment, or revising it to reflect technological progress. This type of manual is recommended if the source code is to be made available.

6. Items Common to All Documents

6.1 *Program Identification:*

6.1.1 Provide the name of the program or model, a descriptive title, and any information necessary to define the version uniquely.

6.1.2 Identify any acronyms or short titles for name of the model.

6.1.3 Note any legal restrictions on use and reproduction.

6.1.4 Describe any relationships to other models.

6.2 *Changes in the Program:*

6.2.1 Provide the name, full identification, and version of the program to be changed.

6.2.2 Identify the equivalent version of the program, with the changes made.

6.2.3 Identify the section(s) changed, and provide the reason(s) for the changes.

6.3 *Authors and Responsibility for Assistance:*

6.3.1 Provide instructions for obtaining more detailed information, or include the position, title, name, telephone number, and mailing address of the individual responsible for providing assistance.

6.3.2 Describe the history of the model's development and the names and addresses of the individual(s) and organization(s) responsible.

6.3.3 Identify current location(s) of the model.

6.4 *Available Material*—List the contents and costs of any program package and the procedure for obtaining this material.

6.5 *Computer Software Abstract*—Summarize the capabilities of the program and the minimum hardware requirements for implementation. ANSI/X3.88 provides additional guide-

lines on the contents of computer program abstracts.

7. Contents of the Technical Document

7.1 *Problem or Function:*

7.1.1 Define the fire problem modeled or function performed by the program, for example, calculation of fire growth, smoke spread, people movement, etc.

7.1.2 Describe the total fire problem environment. General block or flow diagrams may be included here.

7.1.3 Include any desirable background information, such as feasibility studies or justification statements.

7.2 *Technical Description:*

7.2.1 Convey a thorough understanding of the theoretical and mathematical foundations, referencing the open literature where appropriate.

7.2.2 *Theoretical Foundation:*

7.2.2.1 Describe the theoretical basis of the phenomenon and the physical laws on which the model is based.

7.2.2.2 Present the governing equations and the mathematical model employed.

7.2.2.3 Identify the major assumptions on which the fire model is based and any simplifying assumptions.

7.2.3 *Mathematical Foundation:*

7.2.3.1 Describe the mathematical techniques, procedures, and computational algorithms employed to obtain numerical solutions.

7.2.3.2 Provide references to the algorithms and numerical techniques.

7.2.3.3 Present the mathematical equations in conventional terminology and show how they are implemented in the code.

7.2.3.4 Discuss the precision of the results obtained by important algorithms and any known dependence on the particular computer facility.

7.2.3.5 For iterative solutions, discuss the use and interpretation of convergence tests, and recommend a range of values for convergence criteria. For probabilistic solutions, discuss the precision of the results having a statistical variance.

7.2.3.6 Identify the limitations of the model based on the algorithms and numerical techniques.

7.3 *Program Description:*

7.3.1 Describe the program.

7.3.2 List any auxiliary programs or external data files required for utilization of this program.

7.4 *Data Libraries*—Provide background information on the source, contents, and use of data libraries.

7.5 *Validation*—Provide the results of efforts to evaluate the predictive capabilities of the model for a specific use, employing the methodologies outlines in the Guide E 1355.

8. Contents of the User's Manual

8.1 *Technical Document*—Include or summarize the technical document (Section 7).

8.2 *Program Description:*

8.2.1 Include a comprehensive self-contained description of the program.

8.2.2 Define the basic processing tasks performed, and describe the methods and procedures employed. A schematic display of the flow of the calculations is useful.

8.2.3 On-line information (prompts and helps, etc.) can

supplement a printed user manual.

8.3 *Operating and Installation Information:*

8.3.1 Provide instructions for installing the program in the target system. If appropriate, include examples of typical dialogue with the system and test data.

8.3.2 Identify the computer(s) on which the program has been executed successfully and any required peripherals, including memory requirements and tapes.

8.3.3 Identify the programming languages and versions in use.

8.3.4 Identify the software operating system and versions in use, including library routines.

8.4 *Program Considerations:*

8.4.1 Describe the function of each major option available for solving various problems, pay special attention to the effects of combinations of options.

8.4.2 Describe alternate paths that may be dynamically selected by the program from tests on calculated results.

8.4.3 Describe the relationship between input and output items for programs that reformat information.

8.4.4 Describe the method and technical basis for decisions in programs that perform logical operations.

8.4.5 Describe the basis for the operations that occur in the program.

8.5 *Input Data:*

8.5.1 *General Considerations:*

8.5.1.1 Describe the source of input information, for example, handbooks, journals, research reports, standard tests, experiments, etc.

8.5.1.2 Describe special input techniques and requirements, for example, format, blank field treatment, order of items, and field delineation.

8.5.1.3 Describe the handling of consecutive cases. Give the conditions of data retention or reinitialization for the next case.

8.5.1.4 Provide the default values or the general conventions governing those values.

8.5.1.5 Identify the limits on input based on stability, accuracy, and practicality, as well as their resulting limitations to output.

8.5.1.6 When property values are defined within the program, list the properties and the assigned values.

8.5.2 *Specific Considerations for Each Input Variable:*

8.5.2.1 Provide the name of the variable.

8.5.2.2 Give a description or definition.

8.5.2.3 Give the dimensional units.

8.5.2.4 Give the default value, if appropriate.

8.5.2.5 Give the source, if not widely available.

8.6 *External Data Files:*

8.6.1 Outline the general contents and organization of each external data file.

8.6.2 Relate the usage of data files to the execution of the program.

8.6.3 Reference available auxiliary programs that create, modify, or edit these files.

8.7 *System Control Requirements:*

8.7.1 Describe the procedure required to set up and run the computer program.

8.7.1.1 List the operating system control commands re-

quired to execute the program.

8.7.1.2 Include a complete set of the program's prompts, with the ranges of appropriate responses.

8.7.2 Describe how the inputs interact with data files.

8.7.3 Describe how to interrupt the program.

8.7.3.1 For each stage in the program (input, execution, and output), describe how to perform the following functions:

(1) Temporarily halt the program, then resume; and

(2) Halt and exit from the program.

8.7.3.2 Describe the status of files and data after the interruption.

8.8 *Output Information:*

8.8.1 Describe the program output.

8.8.2 Relate the edited output to input options.

8.8.3 Relate the output to appropriate equations.

8.8.4 Describe any normalization of results and list associated dimensional units.

8.8.5 Identify any special forms of output, for example, graphics display and plots.

8.9 *Personnel and Program Requirements:*

8.9.1 State the typical personnel time and set-up time to perform a typical run.

8.9.2 Identify the types of skills required to execute typical runs.

8.9.3 Provide information to enable a user to estimate the computer execution time on applicable computer systems for a typical application.

8.10 *Sample Problems:*

8.10.1 Provide sample data files with associated outputs, to allow the user to verify the correct operation of the program.

8.10.2 Describe the physical problem and associated data files.

8.10.3 Consider the following factors in selecting sample problems:

(1) Choose a benchmark problem or a well-defined example;

(2) Exercise a large portion of the available programmed options; and

(3) Use only a reasonable amount of computer time.

8.10.4 Include the following information in presenting the edited output:

(1) The results of key items in concise forms;

(2) The precision of the results; and

(3) The output parameters, especially the relevance of the order of magnitude of the output.

8.10.5 Provide an order of magnitude of computer execution time for the sample problems, including central processor time, peripheral processor time, and elapsed (clock) time.

8.11 *Restrictions and Limitations:*

8.11.1 List hardware and software restrictions.

8.11.2 Provide data ranges and capacities.

8.11.3 Describe the program behavior when restrictions are violated, and describe recovery procedures.

8.11.4 If accuracy characteristics are significant, describe them in detail.

8.12 *Error messages:*

8.12.1 List instructions for appropriate action when error messages occur.

8.12.2 Describe the manner in which error messages are displayed and explained.

8.13 *References*—List the publications and other reference materials directly related to the fire model or software.

9. Contents of the Installation, Maintenance, and Programming Manual

9.1 *General:*

9.1.1 Reference may be made to appropriate items described in the user's manual. Provide further information, as necessary, to explain the programming details.

9.1.2 Documentation generated by the computer can complement or replace traditional documentation. Examples are a listing of the source program that contains carefully composed comments, a cross-reference dictionary of subroutine names and entry points, and flowcharts of the program logic.

9.2 *System Requirements:*

9.2.1 *Hardware Requirements:*

9.2.1.1 List the machine configuration(s) on which the program has been tested successfully.

9.2.1.2 Enumerate the main memory storage requirements, the amount and type of auxiliary storage (disk and tapes), and the peripheral equipment (printer and plotter).

9.2.1.3 Identify any special hardware, for example, clock and on-line communication channel.

9.2.2 *Software Requirements:*

9.2.2.1 Identify the operating system, language processors, associated subroutine libraries, and supporting programs invoked by the program. Cite the manufacturer's appropriate versions and releases.

9.2.2.2 Describe any known deviations from the manufacturer's supported software that are required by the program, for example, local mathematical and utility routines and other installation-dependent software.

9.3 *Software Structure:*

9.3.1 For proprietary programs or turn-key systems, this documentation may not be available to, or desired by, the user.

9.3.2 *Source Program:*

9.3.2.1 Identify the source language(s).

9.3.2.2 Include a flowchart showing the overall program structure and logic, and detailed flowcharts, where appropriate. The subprogram names should be included on these charts.

9.3.2.3 Pinpoint any known areas of dependency on the local computer installation support facilities.

9.3.2.4 Include a detailed narrative and graphical description of the programming techniques used in writing the program, that is, calling sequence, overlay structure, test plan, common usage, etc.

9.3.2.5 Provide a source listing, or make sure it is readily available.

9.3.2.6 Use comments within the program. The liberal use of comments is a key to understandable programs. An alternative is a commentary keyed to the executable statements of the program.

9.3.3 *Documentation of Subdivisions:*

9.3.3.1 Provide documentation for each major functional subdivision of the program. Such documentation may consist of comments in the program or text explaining the program, or the equivalent.

9.3.3.2 Major functional subdivision includes, but is not limited to, functions, subroutines, loops, and individual subdivisions dependent on decision points.

9.3.4 *Program and Subprogram Details:*

9.3.4.1 Define the role and function of the main program and each subprogram. Identify argument lists and their use.

9.3.4.2 For a particular subprogram, list those routines that call it and, in turn, those subprograms that it may call.

9.3.4.3 Relate the problem variables and constants to the program mnemonics.

9.3.4.4 Describe shared storage assignments, for example, COMMON in FORTRAN.

9.3.4.5 Describe functions performed by machine-dependent subprograms that are unique to the program.

9.3.4.6 Document in detail any subprogram or program module that has a potential of being used by future programs. If documented as a separate entity, it can be referenced or included in the main program documentation.

9.3.5 *Programming Considerations:*

9.3.5.1 Describe the storage allocation and data management procedures. Identify the problem-dependent nature of the memory requirements. Discuss program alternative that affect data storage and the use of data buffering, for example, variable dimensions.

9.3.5.2 Document any overlay or segmentation scheme.

9.3.5.3 Describe the restart, recovery, and successive case capability.

9.3.6 *List of Variables:*

9.3.6.1 List the program and subprogram variables and parameters. The list should include their use and purpose within the program, as well as in its inputs and results. Identify them as local or global variables; that is, do they apply within the module, or are they common to two or more modules of the system?

9.3.6.2 Define all meaningful symbols and arrays used in the routine. Refer to the mathematical or technical notations and terms used in the technical document. Provide units, where applicable. Describe the nominal and initial values of parameters (for example, a computational zero, step sizes, and convergence factors), along with their ranges. Discuss how they affect the computational process.

9.4 *Data Files:*

9.4.1 Specify the names, usage (input, output, or scratch), structure, mode, and data elements of the temporary and external data files.

9.4.2 Discuss program procedures related to the use and maintenance of data libraries and files. Provide the data file retention and allocation requirements.

9.4.3 Enumerate the logical devices employed. Describe the use of each device and any associated data blocking schemes. Identify the contents and format of the information resident on each device. Discuss related physical device use and requirements.

9.5 *External Considerations*—For a program developed as one of a set of programs or as a module in a larger system, provide any constraints and data requirements associated with

incorporating the program in the larger system.

9.6 *Compiling, Interpreting, Assembling, and Loading*—Provide instructions for compiling, interpreting, assembling, and loading the program. If a certain loading sequence is preferred, the reason for such a preference should be given and explained.

10. Keywords

10.1 algorithm; computer program; computer software; documentation; fire model; manual

APPENDIX

(Nonmandatory Information)

X1. COMMENTARY

X1.1 *Background Information:*

X1.1.1 When ASTM Subcommittee E05.39 on Fire Modeling was formed in 1985, one of its mandates, as formulated in response to the results of a survey of ASTM Committee E-5 members and subsequently reflected in the committee's scope, was "to establish a standard procedure for fire model documentation." Task Group E05.39.04 was established to address this mandate and prepare this guide.

X1.1.2 At the time that this guide was prepared, there was no currently accepted documentation standard for computer software specifically for fire models. Fire models released over the previous decade included both well documented and very poorly documented versions.

X1.1.3 This guide is one of several guides being prepared by Subcommittee E05.39 on Fire Modeling. While other standards exist for documentation of computer software (ANSI/ANS 10.2 and 10.3 and ANSI/IEEE 1063), documentation was of such importance to the overall activity of the subcommittee that reference to other standards was not considered adequate.