DOCUMENT RESUME

ED 096 103 SE 016 817

TITLE CACHE Guidelines for Large-Scale Computer Programs.
INSTITUTION National Academy of Engineering, Washington, D.C. Commission on Education.
PUB DATE Feb 73
NOTE 19p.

EDRS PRICE MF-$0.75 HC-$1.50 PLUS POSTAGE
DESCRIPTORS *Chemistry; *Computer Assisted Instruction; Computer Oriented Programs; Computer Science Education; *Engineering Education; *Guidelines; Guides; Science Education
IDENTIFIERS CACHE; *Computer Aids for Chemical Engineering Education

ABSTRACT

The Computer Aids for Chemical Engineering Education (CACHE) guidelines identify desirable features of large-scale computer programs including running cost and running-time limit. Also discussed are programming standards, documentation, program installation, system requirements, program testing, and program distribution. Lists of types of large-scale programs are included. (EB)

ED 096103

CACHE GUIDELINES FOR LARGE-SCALE COMPUTER PROGRAMS

February 1973

PURPOSE OF THE CACHE COMMITTEE

The CACHE (Computer Aids for Chemical Engineering Education) Committee was established by the National Academy of Engineering's Commission on Education to accelerate and coordinate the introduction of digital computation in chemical engineering education. Members of the committee are all chemical engineering educators drawn from the faculties of U.S. and Canadian universities.

MEMBERS OF THE CACHE COMMITTEE

Chairman

Warren D. Seider University of Pennsylvania

Vice-Chairman

Lawrence B. Evans, Massachusetts Institute of Technology

Secretary

Arthur W. Westerberg, University of Florida

Members

Brice Carnahan, University of Michigan (on leave)
James H. Christensen, University of Oklahoma
Eugene Elzy, Oregon State University
Edward A. Grens, University of California, Berkeley
Ernest J. Henley, University of Houston
David M. Himmelblau, University of Texas
Richard R. Hughes, University of Wisconsin
Robert V. Jelinek, State University of New York
A. I. Johnson, University of Western Ontario
Rodolphe L. Motard, University of Houston
Gary J. Powers, Massachusetts Institute of Technology
J. D. Seader, University of Utah
Paul T. Shannon, Dartmouth College
Cecil L. Smith, Louisiana State University
Robert E. Weaver, Tulane University
Imre Zwiebel, Worcester Polytechnic Institute

Ex-Officio Members

Newman A. Hall, U.S. Aid
Matthew J. Reilly, Carnegie-Mellon University (on leave)

Advisory Committee

Donald L. Katz, University of Michigan
John J. McKetta, University of Texas at Austin
W. Robert Marshall, Jr., University of Wisconsin

COMMUNICATION WITH THE CACHE COMMITTEE

The committee actively solicits the participation by interested individuals in the work of on-going CACHE projects. Anyone who wishes to learn more about current CACHE activities may contact any member of the committee or notify The Commission on Education, National Academy of Engineering, 2101 Constitution Avenue, N.W., Washington, D.C. 20418 (Telephone 202—961-1417).

The work of the CACHE Committee is supported in part by the National Science Foundation under Contract C310, T.O. 196.

LARGE-SCALE SYSTEMS TASK FORCE

J. D. Seader, University of Utah, Chairman
L. B. Evans, Massachusetts Institute of Technology
R. R. Hughes, University of Wisconsin
W. D. Seider, University of Pennsylvania
P. T. Shannon, Dartmouth College

CACHE GUIDELINES
FOR
LARGE-SCALE COMPUTER PROGRAMS

TABLE OF CONTENTS

1. INTRODUCTION

In September, 1971, the CACHE Committee issued the document, "Standards for CACHE FØRTRAN Computer Programs." The purpose of that document was to set forth standards governing programming, documentation, testing, and distribution for FØRTRAN computer programs to be made available through the CACHE Committee to chemical-engineering educators. A May, 1972, revision of the standards included a sample program to illustrate the programming and documentation standards. Approximately 100 computer programs that conform to these standards will soon be published and distributed by the CACHE Committee.

In general, the CACHE standards presented in the aforementioned document apply to individual self-standing programs or subroutines that can be implemented readily by the user at his own computing center. However, large-scale or complex systems of programs often require special programming techniques for ready use or efficient operation. Frequently, these systems are best installed and maintained at one computer center, with remote access provided for users elsewhere. Such systems, which are outside of the scope of the former CACHE standards, are defined herein as "large-scale programs."

A number of large-scale computer programs are of definite interest in chemical-engineering education; for example,

a. computer-aided process-design and simulation programs that perform material and energy balances, equipment sizing, and/or economic evaluations for steady-state, continuous chemical processes;

b. component and mixture physical-property calculation programs;

c. some complex equipment-design programs;

d. dynamic process-simulation programs, which simulate process startup, shutdown, and other unsteady operations;

e. computer-aided process-synthesis programs, which develop process flow diagrams, including equipment selection and arrangement;

f. specialized process system programs for pollution control, energy management, etc.

While such programs are available and are in use at a number of universities, the widespread implementation of such programs has been hindered because they are generally difficult to install and maintain on university computers. In order to utilize the programs, an educator has been required to have considerable programming and systems experience and to be agreeable to spending a great deal of time, first in modifying the program to make it function properly on his university's particular computer and then in maintaining it. An alternative procedure for utilizing large-scale computer programs is to install and maintain each program at only one computer, with access by others via communications networks. The present document is concerned with this alternative. Included are discussions of desirable features of large-scale programs, programming standards, documentation, program-installation and system requirements, program testing, and program distribution. It is intended that, within the near future, a limited number of large-scale programs that conform to the standards presented here will be selected by the CACHE Committee for use by chemical-engineering educators.

2. DESIRABLE FEATURES OF LARGE-SCALE PROGRAMS

Before programming standards are considered, it is of interest to summarize desirable features of large-scale programs. Without due consideration to this topic, the complexity normally associated with large-scale programs may limit their application even though the barriers to program access have been essentially removed.

2.1 Running Cost and Running-time Limit

In an academic environment, the cost of running the program may be the major factor. Consequently, it is well to establish some benchmarks for the cost of running the program. In general, the charge per run will be dependent upon the size of the computer, the number of instructions, the iterative nature of the algorithm(s), the amount of input and output, the nature of the mathematical functions utilized, the core space required, the disk file or other storage needed, etc. Because of this complex dependency, it is not possible to determine a simple quantitative relationship between the run charge and the factors listed above. Therefore, it must suffice to state simply that the charge per run should be in the range of approximately one to ten dollars, where the lower figure applies to programs that would usually be run several times during a course by students and the higher figure applies to programs that need only be run once or twice during a course or could be run by groups of students with little loss of instructional effectiveness.

In order to protect the user, the program should have a built-in run-termination feature in the event that execution time exceeds a limit specified as input by the user. Pertinent results computed prior to such a termination should be printed.

2.2 Program Access

Experience indicates that ease of program access from a teletype or other remote job entry (RJE) device can vary widely. Two factors are involved: (1) the commands used to call the program and (2) the manner in which the program is stored and entered into the computer memory. The commands used to call the program from an RJE device should be as simple and as brief as possible. For example, the system commands required before typing in the input data and then for calling and executing a computer-aided process-design program could simply be:

EDIT NEW DATI SAVE	before entering data into data file DATI
OLD CAPD RUN	to call and execute program CAPD with data in DATI

In contrast, some systems are known to require as many as ten commands to access and use a program.

The most desirable method of program storage is a high-speed, high-word-capacity on-line device, because it can rapidly and automatically load the program into core without requiring operator intervention. If the use of such a device is not possible, however, then the program must be stored on a removable device, which would require mounting by an operator. Tapes are acceptable but undesirable.

2.3 Data Input

With large-scale computer programs, the amount of input data required can be substantial; and the need is great to minimize errors. For this reason, the use of free-format input is highly desirable. The input should be as compact as possible, free of redundancy; and, where-ever possible, default values should be provided and clearly defined to the user. Another advantage of free-format input is that it essentially eliminates the need for special input-data sheets for each program.

At least two types of free-format input are in common use. The first type, referred to here as "NØNAME," consists of the values of several variables on each line, with each variable separated by a comma and without regard to any particular field width. Delimiters are utilized to signal the end of a group or subgroup of data values. For example, the following is a line of data: 10, 240., 0.,,, 4., 1.;

The second free-format input scheme, referred to here as the "NAMELIST" type, is based on the use of names for groups of data values in a manner similar to the use of NAMELIST in FØRTRAN. Each line of data input is preceded by the data name. Groups of data values are ordered. Again, the input values are separated by commas and closely spaced. Individual elements of data arrays may also be initialized. An example of a line of this type of input is as follows, with slightly different forms also possible: CØNTRL 1,.0001,20,0,1,10,0.

Both types of free-format input are acceptable. However, the NAME-LIST type is preferable because the data groups may be entered usually without respect to any particular order and because no terminal delimiters may be required. Furthermore, the input-data values are readily identified because they are associated with a group name. Regardless of the type of free-format input used, each line of data should be identified by a sequence number that precedes the data list. Data entry for multiple cases may be by exception.

2.4 Program Output

Printed output from a large-scale computer program should include all user-entered input data, as well as the results of calculations. In many cases, an option should be provided such that the user can obtain either a very limited amount of output or various degrees of more detailed output that traces program execution. Other options should be provided to allow the user to examine any data or data files accessed by the program. Except for debugging purposes, output must not consist of unlabled values. Furthermore, the use of general labels should be avoided. When quantities carry units, these should be attached to the label. Thus, for example,

```
        HEAT EXCHANGER AREA, SQ. FT.        2500.
```

is much preferable to

```
        PARAMETER 5                         2500.
```

In addition, when chemical components are involved, the output should use generally accepted names or abbreviations; *e.g.*, those that appear in the *Handbook of Chemistry and Physics*. Thus, for example,

CØMPØNENT	MØLE FRACTIØN
METHANE	0.1532

is much preferable to

CØMPØNENT	MØLE FRACTIØN
1	0.1532

2.5 Error Messages

An important feature of a large-scale computer program is the use of error messages. They should be incorporated extensively into the program to guide the user in the event that the program runs into difficulty during execution. The error messages should be explicit, intelligible to the user, and flagged sufficiently to draw the attention of the user.

Every attempt should be made by the program to check the input data for obvious irregularities. When detected, an error message should be printed. If the program is to be executed directly or shortly after data entry, the user should be given the opportunity to check and correct the particular line(s) of input data before execution proceeds.

Other types of errors detected during execution may be followed by one of two courses of action after the error message has appeared. If a serious error is detected, the program may be caused to terminate. If the error is not too serious, the program may be allowed to continue. Examples of each of these types of error messages are:

```
**IN INPUT DATA, NUMBER ØF SPECIFIED STAGES IS ØUTSIDE
  ØF PRØGRAM LIMITS.  REVISE DATA AND RESTART.

**BUBBLE-PØINT CALCULATIØN NØT CØNVERGED AFTER LIMIT ØF
  15 ITERATIØNS.  CALCULATIØNS ALLØWED TØ PRØCEED WITH
  LAST TEMPERATURE CALCULATED.

**CØMPUTED TEMPERATURE DRIVING FØRCE IS NEGATIVE.
  SPECIFICATIØNS MAY BE IMPRØPER.  PRØGRAM TERMINATED.
```

Error messages may also include a code number that references a manual, where a more detailed explanation is offered.

In summary, the above-discussed desirable programming features of large-scale computer programs are all aimed at reducing the time, effort, and expense required of the user. Every effort should be made to prepare programs which meet these objectives.

3. PROGRAMMING STANDARDS

Large-scale computer programs need not necessarily be coded in FØRTRAN but, wherever possible, should use FØRTRAN and other high-level languages that are well maintained for the host computer. When FØRTRAN is utilized, USA Standard FØRTRAN, as described in USAS X 3.9-1966 and discussed in the CACHE standards,[1] is required, with the following modifications:

a. Stable, well-established language improvements--*e.g.*, mixed-mode expressions (see Section 2.3 of standards[1])--on the host computer are acceptable.

b. The program may exceed 32K words of machine instructions or single-precision floating-point numbers (Section 2.5[1]).

c. The magnitude of constants is restricted only by the host computer (Section 2.6[1]).

d. Hardware limitations on the host computer do not apply. However, all large-scale systems must be accessible to low-speed terminals.

e. An option must be provided to limit printed output from the program to 70 columns[2] so that teletype output can be conveniently used (Section 2.7[1]).

f. Functions used by the program may include any functions regularly maintained on the host computer (section 2.81[1]).

Other modifications may be permissible, particularly with respect to programming practice (section 2.6[1]), as long as the program is executable on the particular host computer to be employed.

---

[1] *Standards for CACHE FORTRAN Computer Programs* (Washington, D. C.: Commission on Education, National Academy of Engineering, May, 1972).

[2] For interactive programs usable with alphanumeric video terminals, the limiting option should be 50 columns.

When a language other than FØRTRAN is employed, it is essential only that the programming conform to good professional practice, thereby permitting it to be compiled and executed on the particular host computer to be employed. Such special cases must be worked out individually with the CACHE Committee.

4. DOCUMENTATION

The documentation for large-scale computer programs will consist of two separate parts, an Introductory Description and a User's Guide. In certain cases, a Systems Guide may also be necessary.

4.1 Introductory Description

The Introductory Description will be similar to that described in *Standards for CACHE FORTRAN Computer Programs*. It will consist of the following sections:

a. Standard Title Page

b. Technical Description

c. Educational Aspects

The standard title page, presented on p. 10, may omit certain items of program development that are either confidential or not pertinent.

The technical description should be brief and would be best presented by examples of the types of problems that the program is designed to solve. The main purpose of this section is to present just enough description that a potential user can decide whether the particular program is applicable to the problem that he needs to solve. Mathematical and computational details should not be presented here.

CACHE LARGE-SCALE COMPUTER PROGRAM

Program Name:______________________________ **CACHE Number:**________

Language(s):______________________________ **Date Issued:**________

1. **Title:** __(1 line)________________________________
2. **Author(s):**______________________________________
3. **Cognizant Supervisor:**______________________________
4. **Organization Name, Address, and Telephone Number:**________________

   ____

   ____
5. **Abstract: (20-line maximum)**

6. Host Computer and Location:______________________________
7. Remote Terminals Supported: __________________________
8. CACHE Representative and Telephone No.:______________________
9. Host Computer Representative and Telephone No.:________________

   ____

The section on educational aspects should follow the outline given in Section 3.3 of *Standards for CACHE FORTRAN Computer Programs*.

4.2 User's Guide

The User's Guide should provide a description of the engineering, mathematical, and logical methods used, together with step-by-step instructions describing how to use the program. A graded set of sample problems with solutions should be included. The following is a suggested outline for the User's Guide:

1. Summary
2. Introduction
3. General Program Description
4. Program Restrictions and Constraints
5. Preparation of Input Data
6. Description of Program Output
7. Error Messages and Actions
8. Program and Subprograms (identify basic equations and methods of calculation, with reference to pertinent literature)
9. Method of Adding Modules or Subroutines*
10. Sample Problems

The CACHE Committee has on file examples of User's Guides that can be made available. In some cases, the CACHE Committee itself may prefer to assist in the preparation of such a guide.

4.3 Systems Guide

In addition to the type of documentation described above, a list of computing-system requirements must be submitted. A systems manual

*This must not allow the user to change the system.

may be required for programs that need a certain degree of sophistication with regard to their implementation at the host computer. It is recommended that the manual, if required, be prepared with the assistance of the CACHE Committee.

5. PROGRAM INSTALLATION AND MAINTENANCE

The installation of a large-scale computer program on a host computer will be a cooperative effort among the program supplier, a CACHE Committee representative, and a member of the staff of the host computer center.

Initially, during Stage 1 of CACHE Committee operation of large-scale computer programs, program maintenance will be the responsibility of the program supplier. The supplier may be permitted to update the program after consultation with the CACHE representative and after notification of all program users. The program supplier will be responsible for providing modifications of the appropriate documentation to the CACHE Committee representative.

Eventually, program maintenance will be the responsibility of the program supplier, the CACHE maintenance corps, and the staff of the host computer center. Where the level of usage is great, the staff of the host computer center will be encouraged to provide application program support.

Normally, routine maintenance of the program will consist of checking the computer-center program file once each month and running a sample problem at least once every six months. This will be the responsibility of the CACHE Committee representative.

6. CONSULTATION SERVICES

In general, it is anticipated that the large-scale programs will be sufficiently tested and documented that supplier consultation with users will not normally be necessary. The following refers mainly to procedures for handling errors.

Large-scale-program users will refer errors suspected to be associated with the operating system of the host computer to the staff of the host computer center. A computer "mail box" will be provided at the host computer.

Initially, problems associated with the execution of the large-scale program will be reported to the program supplier. It is important that program suppliers be willing to answer questions on a regular basis. Eventually, these problems will be reported first to the CACHE maintenance corps and then to the program supplier.

7. PROGRAM TESTING

Before final acceptance by the CACHE Committee, a candidate large-scale computer program will receive a comprehensive check-out according to the following procedure. This should ensure success when the program is subsequently accessed by users.

a. The program supplier will run sample problems at his computer center.

b. A CACHE Committee representative will run the same sample problem after the program has been installed at the host computer.

c. The CACHE Committee representative will run a new problem that he devises.

d. Another CACHE Committee member or associate will run a sample problem by RJE from a different location.

The basic objective is to achieve program use without interaction with the program supplier. However, if difficulties arise in carrying out this procedure, it may be necessary to contact the program supplier.

8. PROGRAM REVIEW AND DISTRIBUTION

A representative of the CACHE Committee will serve as the program reviewer and documentation editor for the large-scale computer programs. He will seek the assistance of other CACHE members and associates in carrying out this function. Submission, review, and approval of each candidate program will occur in several distinct steps, characterized by the documentation required:

a. a title page containing the program abstract in the form shown on page 10;

b. documentation for the Introductory Description, as described earlier, including Technical Description and Educational Aspects, together with a preliminary Program Description and a sample problem with input and printed output;

c. a draft of the User's Guide, as described above;

d. a draft of the systems manual, if required;

e. installation of the program on the host computer;

f. a four-step check-out of the program, as outlined above;

g. the final version of the User's Guide.

At each of the above steps, candidate programs will receive careful review to make sure that they meet the standards discussed above. Careful attention will be paid to those desirable features discussed earlier. Potential program suppliers may be requested to modify their programs or write-ups to gain acceptance.

It is planned that program booklets will be published as required, which will contain the Introductory Descriptions for several large-scale programs. The User's Guides will be published separately. It is anticipated that the program booklets will receive wide distribution, while distribution of the User's Guides will, in general, be restricted to serious users who request copies.

9. LIST OF TYPES OF LARGE-SCALE PROGRAMS

A major goal of the CACHE Committee is to achieve greater involvement of chemical-engineering faculty in problem solving by means of the digital computer. For this reason, it is believed that large-scale programs for adoption by the CACHE Committee should be selected to cover most areas of undergraduate and graduate education in chemical engineering. A list of possible program types that would meet this criterion is given in Table I. While other types of programs will be considered, those listed are particularly desired. In addition, candidate programs should be such that students can learn to use them in a reasonable period of time. In general, programs that are likely to be used only by the instructor for demonstration purposes will not be given serious consideration.

TABLE I: POSSIBLE TYPES OF PROGRAMS FOR INSTALLATION ON NETWORKS

PROGRAM AREA	*EXAMPLE*
Material and energy balances	Material and energy balances for steady-state processes
Thermodynamics	Thermodynamic & other physical properties
Equilibrium-stage calculations	Equilibrium flash with necessary data base
Equilibrium-stage calculations	Complex distillation package
Heat transfer	Heat-exchanger design
Mass Transfer	Design of a nonisothermal packed absorber
Fluid mechanics	Piping network
Kinetics	Adiabatic plug-flow reactor
Kinetics	Fixed-bed catalytic reactor
Process design	Steady-state process design or simulation
Process design	Dynamic simulation
Process synthesis	Process synthesis
Optimization	Multivariable constrained optimization
Optimization	Multivariable unconstrained optimization
Pollution control	Water-treatment-plant design
Pollution control	Environmental games