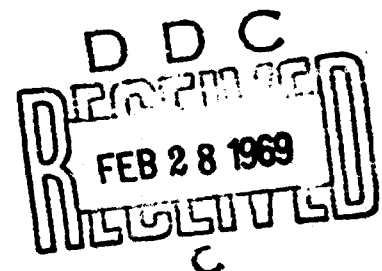# Research Computation Center Program Publication Guide

DORIS E. GOSSETT, JANET P. MASON, AND ALAN B. BLIGH

Research Computation Center
Mathematics and Information Sciences Division

AD 682924

February 1969

DDC
FEB 28 1969
C

NAVAL RESEARCH LABORATORY
Washington, D.C.

CONTENTS

ABSTRACT

A guide has been developed that contains the Research Computation Center standards for documenting, revising, classifying, submitting, identifying, obtaining, and evaluating computer programs at NRL. It is anticipated that the use of such standards will promote efficient organization of an RCC Program Library and will provide for maximum effective usage of this program library throughout NRL.

PROBLEM STATUS

A final report on one phase of the problem; work is continuing on other phases.

AUTHORIZATION

ACKNOWLEDGMENT

ii

RESEARCH COMPUTATION CENTER PROGRAM PUBLICATION GUIDE

I. INTRODUCTION

This guide contains the Research Computation Center (RCC) standards for documenting, revising, classifying, submitting, identifying, obtaining, and evaluating programs which exist or are to exist in the Control Data Corporation CO-OP Users' Group (CO-OP) and/or RCC Program Library. These proposed standards are in part patterned after those proposed by CO-OP.

Page numbers of this Guide consist of the Chapter Number followed by a sequential number; e.g., "2-1" means Chapter II, page 1.

Chapter II contains information on how to specify zero and the letter O, how to specify blanks, and when to use upper case and lower case letters in documentation.

Chapter III specifies the format for program documentation. Write-ups of major programs of a permanent nature will be published as NRL Reports and/or NRL Computing References. Write-ups of other programs that are of a more temporary or changeable nature will be published as NRL Memorandum Reports and/or NRL Computing Bulletins. The decision as to which category a write-up belongs usually will be made by the program originator and the RCC. Programs will also be submitted to CO-OP upon request by the contributor.

Chapter IV describes the program identification numbering system that is used to uniquely identify each program or subroutine deck. This number will also be used for filing purposes in the RCC Program Library.

Chapter V specifies standards for internal documentation of programs using comment or remark cards.

Chapter VI defines the minimum equipment configuration that is assumed in documentation of programs.

Chapter VII specifies the procedures for supplementing or revising a program write-up.

Chapter VIII specifies the procedure for submitting a program to CO-OP and/or RCC for inclusion in the RCC Program Library.

Chapter IX specifies the contents of a program brief which is necessary in order to obtain an RCC Identification Number for a program deck. Addendum b contains a sample program brief.

Chapter X specifies the procedure for obtaining a program which exists in the CO-OP and/or RCC Program Library.

Chapter XI gives the submittal form required by CO-OP when submitting a program. Addendum C contains a sample submittal form.

Chapter XII gives the evaluation form required by the RCC when using a CO-OP and/or RCC Library program.

Chapter XIII lists and defines the classification codes used by CO-OP and the RCC.

Addendum A gives the write-up of a subroutine using the recommended format given in Chapter III.

Addendum B contains a sample program brief as described in Chapter IX.

Addendum C contains a sample submittal form as described in Chapter XI.

It should be noted that throughout this guide, any reference to the RCC Program Library is not a reference to the computer system library (i.e., the on-line library file). The RCC Program Library is simply a collection of programs maintained on file by the RCC, where each program deck is uniquely identified by an RCC Identification Number as defined in Chapter IV.

II. DOCUMENTATION STYLE STANDARDS

GRAPHIC CONVENTIONS

O versus ∅

The letter O is typed ∅ when it appears in examples
and definitions of text to be input to the computer,
but not otherwise. For example: The control card
"$_9^7$C∅MPASS,L" will cause COMPASS assembly with a
listing. In particular, examples of computer output
do not need slashed O's.

Blanks

When it is necessary to call attention to explicit
blanks in a text, carets are used. For example:
The termination label is "∧ENDREEL∧FINAL∧∧".

UPPER CASE versus LOWER CASE

When referring to a language in general it is appropriate
to write the language name in lower case letters with the
initial letter in upper case (e.g. - Fortran). However,
the proper name of a particular compiler or system is
usually written in upper case (e.g. - FORTRAN IV, DRUM
SCOPE).

III. FORMAT FOR DOCUMENTING PROGRAMS AND SUBROUTINES

Shown below is the basic outline of the format, to be used whenever possible, to document programs or subroutines. Following the basic outline is the format including a detailed description of each heading. This format is patterned after that contained in the "CO-OP Handbook, Part III: Program Catalog, Section 1: Program Classification Requirements" of June 1968. Users wishing to submit their programs to the RCC for distribution to CO-OP and/or to other users are encouraged to follow these conventions.

All items of the format should be completed by the author where possible. All information essential to the usage of the routine must be included. When an item is not applicable, or has not been determined, this should be explicitly stated. If necessary, the user may consult with RCC staff members to determine how to handle any items in doubt. Throughout the write-up, examples should be used to clarify and illustrate usage.

Addendum A gives the write-up of a subroutine using the format specified below.

The basic outline is as follows:

1.0 IDENTIFICATION
    1.1 Title
    1.2 Identification Name
    1.3 Classification Code
    1.4 RCC Identification Number
    1.5 Entry Points
    1.6 Programming Language
    1.7 Computer and Configuration
    1.8 Contributor or Programmer
    1.9 Contributing Organization
    1.10 Program Availability
        1.10.1 Submittal
        1.10.2 On File
    1.11 Verification
    1.12 Date

2.0 PURPOSE
    2.1 Description of the Routine
    2.2 Problem Background

3.0 USAGE
    3.1 Calling Sequence or Operational Procedure
    3.2 Arguments, Parameters, and/or Initial Conditions
    3.3 Space Required (Decimal and Octal)
        3.3.1 Unique Space
        3.3.2 Common Storage
        3.3.3 Temporary Storage
    3.4 Messages and Instructions to the Operator

4.0 METHOD OR ALGORITHM

5.0 FLOW CHART AND/OR SOURCE LANGUAGE LISTING

6.0 COMPARISON

7.0 TEST METHOD AND RESULTS

8.0 REMARKS

The format including a description of each item is as follows:

1.0 IDENTIFICATION

1.1 Title

Give a descriptive title of the routine, limited
to approximately fifty characters which are included
in the Fortran character set, to facilitate indexing
and listing.

1.2 Identification Name

This descriptor consists of the following three
items of information, separated by hyphens:

(a) the two character classification code
as defined in section 1.3.

(b) the CO-OP code for the contributing
organization (NRL).

(c) an alphanumeric descriptor of the routine
not to exceed eight characters. This
name can not contain embedded blanks or
special characters.

The identification name is used by CO-OP and the
RCC for further indexing and identification.

For example, given an NRL routine with a classification
code of C3 and a descriptor of BESSEL, the identifi-
cation name would be C3-NRL-BESSEL.

3-2

1.3 Classification Code

Give both the code designation(s) and the verbal translation. The code designation consists of a letter, indicating the primary class, followed by a digit indicating the sub-class. The available classification codes and their descriptions are given in Chapter XIII.

1.4 RCC Identification Number

This eight digit code number uniquely identifies each program or subroutine deck. The content and construction of this number is described in Chapter IV. List each program and subroutine along with the identification number for each when applicable.

1.5 Entry Points

Give the alphanumeric code name of each entry point in alphabetical order. If the name of the routine in which the entry point occurs is different from the entry point name, then the name of the routine should appear in parentheses to the right of the entry point name.

1.6 Programming Language

Indicate the language(s) in which the routine is written, and specify the type of routine, i.e., program, subroutine, or system. Also specify the operating system under which the routine was written and tested.

1.7 Computer and Configuration

Specify the computer for which the routine was written (e.g. - CDC 3800). If a special configuration or equipment complement different from that defined in Chapter VI is needed, state the particulars.

1.8 Contributor or Programmer

Specify the name, NRL code, branch, and division of the programmer(s) who wrote the routine. If someone other than the programmer(s) given should be contacted in case of difficulties with the routine include this information.

1.9   Contributing Organization

Include the CC-OP organization code (NRL), name
(Naval Research Laboratory), and address (Washington,
D.C. 20390) of the contributing organization.

1.10 Program Availability

1.10.1 Submittal

List all of the materials pertaining to the
routine which are being submitted to CO-OP
and/or RCC (Example: Program write-up,
Fortran source deck, source listing, etc.).

1.10.2 On File

Specify means of obtaining the routine if
other than through the means stated in
Chapter X. For example, the routine might
be on the on-line system library, in a perma-
nent file on mass storage, or on file by the
contributor. Also specify all the materials,
pertaining to the routine, which exist on file
by the contributor if other than those given
in Section 1.10.1.

1.11 Verification

This section is to be used by the RCC to indicate to
what extent the RCC has checked the routine and the
testing procedures used.

1.12 Date

Specify the date of the routine write-up. Should any
revisions be made to an existing routine, the original
date and the dates of any other revisions shall be
included together with the date of the current revision.

2.0 PURPOSE

2.1 Description of the Routine

Give a brief general description of what the routine
does. The purpose should provide sufficient information
so that a reader can decide whether or not the routine
may meet a particular need.

2.9 Problem Background

Specify the unique problem characteristics that led
to the program, prior techniques employed in an attempt
to resolve the problem, and limitations of prior
techniques that prevented an effective solution.

3.0 USAGE

3.1 Calling Sequence or Operational Procedure

Specify the call(s) or procedure(s) which is necessary
to utilize the routine(s), including the use of all
entry points. Specify and define the function of all
entry points.

3.2 Arguments, Parameters, and/or Initial Conditions

Describe all arguments, parameters, and arrays needed
to utilize the routine(s). This description should
include what the arguments represent in the physical
problem, mode, and any size information or limitations.
Also describe any assumptions made by each routine
when it is called.

3.3 Space Required (Decimal and Octal)

3.3.1 Unique Storage

Specify all unique core space requirements of
the routine, excluding common blocks.

3.3.2 Common Blocks

Specify the name and length of all common blocks
required by the routine. Also specify the names
of the arrays or cells within each common block,
including their length, their function, their
mode, and all routines which use each cell or
array. One concise way to show some of this
information is shown by the following imaginary
example:

3-5

ROUTINES	COMMON BLOCKS					
	BLOCK RDC				BLOCK SRT	
	ADVG	ADVØ	*NCW	NTP	*SØRTAR(257)	*SCALE(96)
ADDLIST			X	X		X
FI	X	X				
MULC		X		X	X	X
RDSPEC	X		X			
SPECPR		X	X	X		X
SPECPN		X	X			
TOTAL	X				X	

* indicates real (floating point) mode
(all other variables are in integer mode)

3.3.3 Temporary Storage

Temporary storage areas are those areas that
contain pertinent information only at certain
times in a run and at all other times the contents
of the areas could be destroyed without consequence.
Specify the names of temporary storage and the size
of each along with the names of all routines that
use the temporary storage. It should be appropr-
iately noted if the temporary storage is included
in unique or common.

3.4 Messages and Instructions to the Operator

Specify and define any messages given to the operator by
the routine and any special operator instructions such as
special tape handling procedures. In general, output
messages to the operator are discouraged by the RCC.

3.5 Error Returns, Messages, and Codes

Include a list of the error situations (error returns,
error codes, and error stops) that are detected and the
action taken. Any diagnostic messages should be speci-
fied and described. Also specify whether each error
causes normal or abnormal termination of job execution.

3-6

If an error situation in this routine can cause
diagnostic action by another routine or the system
monitor, this action should be described. When
possible, recommend the action for the programmer
and/or operator to take.

3.6   Informative Messages to the User

Specify and define any messages that give information
rather than indicate an error. All messages in this
category assume that the run continues without any
action on the part of the user.

3.7   Input

Include (where applicable) the input media (cards,
magnetic tape, etc.), logical unit assignments,
labels, densities, tape numbers, and clarifying
descriptions for all input required by the routine.

3.8   Output

Include (where applicable) the output media, logical
unit assignments, labels, densities, tape numbers,
and clarifying descriptions for all output generated
by the routine. Estimates of the output volume should
also be included when possible.

3.9   Formats

Describe the formats and contents of all input and
output data. Often it is helpful to include a list
of format statements or format specifications for
each field.

3.10   External Routines and Symbols

List in alphabetic order each routine and the external
symbols that each routine calls. One concise way to
show this information is by use of a table that would
have the names of the routines down the left side, all
external symbols across the top and checks in the
appropriate rows and columns. If the name of the
routine in which the external symbol occurs is differ-
ent from the external symbol, then the name of the
routine should appear in parentheses to the right of
the external symbol. Also indicate where each external
routine is physically located (system library, RCC
Program Library, etc.) if not within the program deck.

3.11 Timing

When possible, estimates of running time should be provided.

3.12 Accuracy

For mathematical routines, statements of accuracy, precision, or the magnitude and propagation of errors should be included.

3.13 Cautions to Users

Include anything of a cautionary nature.

3.14 Program Deck Structure

Specify the deck structure for a run, including control cards, routines, and data.

3.15 References - Literature - Appendices

List and describe any references, literature, appendices, or other documentation that pertain to this routine.

4.0 METHOD OR ALGORITHM

Describe how the routine works in regard to the mathematics or algorithm. Describe the programming techniques and methods, supporting theory, design, computation equations, and the derivations of the computation equations to substantiate or illustrate the program and emphasize its advantages, if any, over previous methods.

5.0 FLOW CHART AND/OR SOURCE LANGUAGE LISTING

Flow charts should be submitted whenever they are available and not too lengthy, particularly for machine language routines. Large programs should have both general and detailed flow charts. The general flow chart consists of one block for each subroutine. A source language listing should only be included when it is fairly short. If a flow chart or source language listing is available but is not being included here, state how it may be obtained.

3-8

6.0 COMPARISON

If similar programs exist in the CO-OP and/or RCC Program
Library a comparison should be made with respect to such
programs discussing the advantages and disadvantages of the
new routine and indicating any programs which are made
obsolete.

7.0 TEST METHOD AND RESULTS

Include test data used and results obtained in checking the
operation and accuracy of the routine.

8.0 REMARKS

Give any other information that may be useful.

IV. PROGRAM IDENTIFICATION NUMBERING SYSTEM

An RCC program identification number is given to any program or subroutine deck to be filed in the RCC Program Library. This number provides a means for identifying the following items:

1. Source program or subroutine decks
2. Application classification of the program
3. The program originator
4. The level of program revision

The RCC program identification number consists of eight characters that are determined as follows:

Character Number	Function	Current Assignments
1,2	Classification Code	One letter followed by one digit (See list in Chapter XIII)
3,4,5	A 3-digit serial (within each class) number assigned by RCC	000-999 (range)
6	Originator code	C - CO-OP R - RCC Ø - Other NRL T - Tenant Activities N - Non-NRL
7,8	A 2-digit program revision code	00-99 (range)

A program identification number can be obtained from the RCC upon submittal of a Program Brief (see Chapter IX) to the RCC by the programmer. If the program deck is to be physically located in the RCC Program Library file it is additionally required that the program card deck which is submitted must

(a) be properly sequence-numbered in card columns 73-80 with sequential numbers increasing in value through the deck, and

(b) contain internal documentation as given in Chapter V.

The RCC program identification number should be included on a comment or remark card following the routine name, at the beginning of the symbolic deck (see Chapter V). The identification number should also be included in any documentation about the routine.

The program revision code is used as follows: The initial version
of a new program or subroutine deck is assigned a revision code of "00."
Each time the program or subroutine is revised or modified in any way, this
number is incremented by one.

V. PROGRAM INTERNAL DOCUMENTATION STANDARDS

Each program or subroutine deck should contain internal documentation in the form of comment or remark cards. This internal documentation should perform three functions:

1. Identify
2. Define usage
3. Clarify program

Identifying comments or remarks are strongly recommended for all programs and are required by the RCC before a routine will be put in the RCC Program Library or accepted for publication. These comments or remarks should appear immediately following the routine name and should contain the following information in the specified order:

        IDENT NUMBER -
        TITLE -
        IDENT NAME -
        LANGUAGE -
        COMPUTER -
        CØNTRIBUTØR -
        ORGANIZATION -
        DATE -
        PURPOSE -

The above headings are defined in Chapter III, Sections 1.0 and 2.0. The information supplied in these comments or remarks need not be as detailed as in the write-up. The above identifying headings should be included as part of the comments or remarks.

Usage information comments or remarks are not required but are strongly encouraged. These comments or remarks should follow the identifying comments or remarks and should include the following type of information:

        CALLING SEQUENCE -
        ARGUMENTS -
        I/O INFORMATION -
        EXTERNAL SYMBOLS -

The above headings are defined in Chapter III, Section 3.0. The information supplied in these comments or remarks need not be as detailed as in the write-up. The above headings should be included as part of the comments or remarks.

Clarifying comments or remarks should appear throughout the routine to describe what the routine is doing at any given point.

In addition to the above documentation, it is required that each deck of a routine be properly sequence-numbered in card columns 73-80 with the numbers increasing in value through the deck.

VI. MINIMUM EQUIPMENT CONFIGURATION

The minimum equipment configuration assumed for all documentation of routines is the basic CDC 3800 computer with 32K core storage, 2 magnetic tape units, drums, card punch, card reader, and a line printer. If a routine requires equipment in addition to that stated above, such should be stated in the documentation (Chapter III, Section 1.7). The above minimum configuration is similar to that given by CO-OP.

VII. PROGRAM ADDENDA AND REVISIONS

When an error is found an RCC/CO-OP Library Evaluation Form (see Chapter XII) should be sent to the RCC Program Library, Code 7813L. This evaluation should explain the error in detail. To correct or add to an existing routine, use the documentation format given in Chapter III and follow the standard program submittal procedure described in Chapter VIII. Only those items that are affected need be resubmitted. Only the originator (unless he is unavailable) of a program can make corrections or modifications to the CO-OP and/or RCC Program Library file copy of a program.

An addendum to a publication will be indicated by a letter following the original publication number (e.g. - A, B, etc., for each succeeding addendum).

A revision of a publication will normally be accomplished by re-publishing the entire document. A revision of a publication will be indicated by .n following the RCC publication number, where n = 1,2,.... for each succeeding revision. However, a revision to an NRL Report or an NRL Memorandum Report will be given a new report number, with the old number specified as being superseded. A revision should specify the dates of the original and all revisions.

VIII. PROGRAM SUBMITTAL PROCEDURE

Users having programs cf general interest which have been thoroughly tested and documented may submit the programs to the RCC for publication as an NRL Computing Note, Bulletin, or Reference (Programs to be sent to CO-OP or to be documented as NRL Reports or Memorandum Reports must follow additional standard procedures.). The programs should be written in a compiler or assembly language generally available through the RCC. A memorandum should be written to the RCC Program Library, Code 7813L requesting that the program be published and included in the RCC Program Library. Included with the memorandum should be the following:

1. Program Brief (defined in Chapter IX; example in Addendum B) if the RCC does not already have one for the routine(s). If a Program Brief is currently in existence, the RCC Identification Number for the routine must be supplied.

2. Program Source Listing resulting from the assembly or compilation of the submitted source deck.

3. Program Source Deck on cards, interpreted, sequence-numbered, and internally documented (see Chapter V); or the RCC Identification Number of the routine(s) if already in existence in the RCC Program Library file.

4. Program write-up preferably in the form specified in Chapter III. See the example in Addendum A.

If a program is to be sent to CO-OP, the CO-OP Submittal/Revision Form must also be supplied (see Chapter XI and Addendum C).

IX. PROGRAM BRIEF

In order to obtain an RCC Identification Number for a program or subroutine deck, a Program Brief must be supplied to the RCC. The Program Brief should contain the following information:

TITLE.

IDENTIFICATION NAME.

PROGRAMMING LANGUAGE.

COMPUTER AND CONFIGURATION.

PROGRAMMER.

NRL ORGANIZATION.

DATE.

PURPOSE.

PROGRAM STATUS.

The above headings are described in pert by Chapter III, Sections 1.0 and 2.0. The NRL ORGANIZATION section should include the NRL Division, Branch, and Code, in that order. The PROGRAM STATUS section should indicate what the current status of the program is (complete, incomplete, etc.).

A sample Program Brief has been prepared in Addendum B.

X. PROCEDURE FOR OBTAINING A PROGRAM

All programs, for which documentation exists in the form specified in Chapter III, will be available in the RCC Program Library unless stated otherwise in the write-up (Chapter III, Section 1.10.2). To obtain a program existing in the CO-OP and/or RCC Program Library contact Research Computation Center Program Library personnel. Users of all routines obtained from the CO-OP and/or RCC Program Library are requested to submit an RCC/CO-OP Library Evaluation Form (see Chapter XIII) to the RCC Program Library, Code 7813L. This evaluation should contain detailed usage information about discovered errors or inconsistencies, and any other pertinent information.

XI. USER GROUP LIBRARY SUBMITTAL/REVISION FORM

The Program Abstract Cover Sheet as given at the end of this chapter must be filled out whenever a program, program revision, or write-up revision is being submitted to CO-OP. A sample Program Abstract Cover Sheet has been prepared in Addendum C. The categories on the form are defined as follows:

1. Library: The word "CO-OP" should be placed in the blank preceding the word "LIBRARY" to indicate that the routine should be placed in the CO-OP Library as opposed to SWAP, POOL, etc.

2. Catalog Identification: The "Identification Name" information as described in Chapter III, Section 1.2.

3. Title (Descriptive): See Chapter III, Section 1.1.

4. Source Language(s) or Dialect(s): Name of specific source language(s) and dialect(s), or versions, in which the program is coded.

5. Operating System and Version: Name of Operating System (e.g., DRUM SCOPE) and Version (e.g., Version 2.0).

6. Computer and Configuration: The computer and configuration for which the program has been written and checked out should be given. This should include the storage space required.

7. Contributing Organization: The full name of the contributing organization as defined in Chapter III, Section 1.9.

8. Submitter/Programmer: The name(s) of the programmer(s) who wrote the program. Include the programmer(s) NRL Code(s).

9. Revisor: The name(s) of the programmer(s) who revised the program. This is left blank when the original program is being submitted.

10. Date Written: The date of the original program shall be specified. When submitting a revision, the date of the revision should also be given.

11. Program Material: The number of pages in the program write-up (including the Program Abstract Cover Sheet) should follow "Write-up Page Count".

If a card deck is submitted, the number of cards in the deck should follow "Sequenced Source-Card Count". If the program is submitted on magnetic tape it should consist of one physical file and the total number of BCD card images should be entered following "M/T Record Count".

If the program is submitted on paper tape, the number of recognizable records on the tape should be entered following "P/T Record Count".

11-1

12. <u>Other Available Documents</u>: The submitter should specify when other items are submitted, such as listings, flow-charts, sample print-outs, etc. A listing need not be included if the card deck is sequenced.

13. <u>Original/Revised Program Catalog Abstract</u>: The submitter should type his abstract exactly as he wishes it to appear in forthcoming issues of user group program catalogs. The abstract should be no longer than two short paragraphs and should contain the following information:

      Purpose
      Entry Points (in alphabetical order)
      Program Type (subroutine, complete routine, or system)

14. <u>Nature of Revision (or Additional Information)</u>: This space is provided for giving more detailed information about a revision than is needed for the abstract in the program catalog.

All information provided on the Program Abstract Cover Sheet must be in standard 48-character code, for keypunching. Therefore, mathematical symbols and other special characters, though permissible elsewhere in the documentation, must be converted for typing on the Program Abstract Cover Sheet.

Copies of the Program Abstract Cover Sheet are available from the RCC.

CATALOG IDENTIFICATION : _____   REVISION :_____

TITLE (DESCRIPTIVE) :   _____

                          _____

SOURCE LANGUAGE(S)
AND DIALECT(S) :   _____

OPERATING SYSTEM
AND VERSION :   _____

COMPUTER & CONFIGURATION : _____

                           _____

CONTRIBUTING ORGANIZATION : _____

SUBMITTOR/PROGRAMMER : _____   REVISOR : _____

DATE WRITTEN :   ORIGINAL _____   REVISION _____

PROGRAM MATERIAL :   WRITE-UP PAGE COUNT : _____

                       SEQUENCED SOURCE-CARD COUNT : _____

                       M/T RECORD COUNT : _____   P/T RECORD COUNT : _____

OTHER AVAILABLE DOCUMENTATION : _____

ORIGINAL / REVISED PROGRAM CATALOG ABSTRACT :

NATURE OF REVISION (OR ADDITIONAL INFORMATION) :

For Completion by Distribution Agency :
     Original/Previous Revision : UGN _____ Page ____ Accessed _____
     Latest Revision :   UGN _____ Page ____ Accessed

XII. RCC/CO-OP LIBRARY EVALUATION FORM

Each user of an RCC or CO-OP Library program is requested to submit
an RCC/CO-OP Library Evaluation Form, as given at the end of this chapter,
to the RCC Program Library, Code 7813L. This evaluation should specify
in detail, any difficulties encountered and any corrective measures taken
while using the program.

RCC/CO-OP LIBRARY EVALUATION

DATE_____

RCC/CO-OP IDENTIFICATION _____

PROGRAM TITLE _____

LANGUAGE _____

USER'S APPLICATION:

DIFFICULTIES ENCOUNTERED, IF ANY:

COMMENTS, IF ANY:

USER'S NAME _____

CODE _____

TEL. EXT _____

PLEASE SEND COMPLETED FORM TO RCC PROGRAM LIBRARY, CODE 7813L.

XIII. CLASSIFICATION CODES

The following classification system is used by the RCC to categorize computer subroutines and programs. This classification system is almost identical to that currently proposed by CO-OP in the "Users Organizations for Control Data Computer Systems Handbook, Part III: Program Catalog, Section 1: Program Classification Requirements", June 1968. The classification code consists of a letter, indicating the primary class, followed by a digit indicating the sub-class. Under each primary class will be a zero code having the heading "General" under which will be filed the routines that are not appropriate for the listed secondary classes.

The primary classes are listed below after which the entire classification system is given including a description of some of the primary and sub-classes.

A. ARITHMETIC ROUTINES
B. ELEMENTARY FUNCTIONS
C. POLYNOMIALS AND SPECIAL FUNCTIONS
D. OPERATIONS ON FUNCTIONS AND SOLUTIONS OF DIFFERENTIAL EQUATIONS
E. INTERPOLATION AND APPROXIMATIONS
F. OPERATIONS ON MATRICES, VECTORS AND SIMULTANEOUS LINEAR EQUATIONS
G. STATISTICAL ANALYSIS AND PROBABILITY
H. OPERATIONS RESEARCH TECHNIQUES, SIMULATION AND MANAGEMENT SCIENCE
I. INPUT
J. OUTPUT
K. INFORMATION TRANSFER
L. EXECUTIVE ROUTINES
M. DATA HANDLING
N. DEBUGGING
O. SIMULATION/EMULATION OF COMPUTERS AND DATA PROCESSING INTERPRETERS
P. DIAGNOSTICS
Q. SERVICE OR HOUSEKEEPING: PROGRAMMING AIDS
R. NON-NUMERICAL MATHEMATICS
S. INFORMATION RETRIEVAL
T. APPLICATIONS AND APPLICATION-ORIENTED PROGRAMS
U. LINGUISTICS AND LANGUAGES
V. GENERAL PURPOSE UTILITY SUBROUTINES
X. DATA REDUCTION
Y. USER ASSIGNED
Z. ALL OTHERS

The entire classification system is as follows:

A. <u>ARITHMETIC ROUTINES</u>

Routines of the form of sub-routines or partial programs to be incorporated into programs or subroutines for the purpose of doing arithmetic operations within a given algebraic structure.

AO. General

A1. Real

Routines which perform exclusively real arithmetic operations, including single precision, extended precision, and multiple precision representations of real numbers (real opposed to imaginary).

A2. Complex

Routines which perform complex arithmetic operations including extended and multiple precision representations on complex numbers.

A3. Byte

Routines which perform single or multiple precision byte arithmetic.

A4. I/O Routines

I/O Routines designed for use with (e.g.) multiple precision arithmetic packages frequently contain entries for both input and output, hence could legitimately be filed under I or J. However, their real intent and purpose is for use only with special routines already filed under A (filing under I or J therefore merely loses them for prospective users). While this type of I/O package is not programmed arithmetic in the true sense of the term, its intimate relationship with A-category routines merits that classification.

B. <u>ELEMENTARY FUNCTIONS</u>

Elementary functions are defined to be those generated by constants and the independent variable by means of a finite sequence of elementary operations. Elementary

operations on functions F(X) and
G(X) are those that yield any of
the following F(X) = G(X), F(X)*
G(X), F(X)/G(X), (F(X))**A,
A**(X), LOG (X), T (X) where T
is any trigonometric or inverse
trigonometric function and X is
any number (fixed-point, floating-
point, complex, multiple-precision,
etc.).

B0.   General

B1.   Trigonometric and
      Inverse Trigonometric

B2.   Hyperbolic and Inverse
      Hyperbolic

B3.   Exponential and Logarithmetic

B4.   Roots and Powers   Refers to roots of quantities
                                not polynomials.

C. POLYNOMIALS AND SPECIAL   Routines which include operations
    FUNCTIONS   on and evaluation of polynomials
                                 and functions often referred to
                                 as special functions, such as
                                 hypergeometric, gamma, bessel,
                                 elliptic, etc. Simultaneous
                                 nonlinear equations shall include
                                 operations on and solutions of
                                 systems of nonlinear algebraic and
                                 transcendental equations but shall
                                 exclude differential type equations.

C0.   General

C1.   Evaluation of Polynomials

C2.   Roots of Polynomials

C3.   Evaluation of Special Functions

C4.   Simultaneous Nonlinear Algebraic Equations

C5.   Simultaneous Transcendental Equations

D. OPERATIONS ON FUNCTIONS AND SOLUTIONS OF DIFFERENTIAL EQUATIONS

This category shall include routines to perform numerical integration and numerical differentiation as well as routines to solve differential equations both ordinary and partial. Other routines which evaluate operations of functions or perform the operation on a function shall also be included.

D0. General

D1. Numerical Integration

D2. Numerical Solutions to Ordinary Differential Equations

D3. Numerical Solutions to Partial Differential Equations

D4. Numerical Differentiation

E. INTERPOLATION AND APPROXI- MATIONS

Interpolation shall include routines which, when given a set of points, approximate values based on polynomials, ratios of polynomials, sines and/or cosines, exponential functions, etc. Approximations shall mean routines to provide functions or curves to fit a set of points or data by polynomials, exponentials, sines, cosines, etc.

E0. General

E1. Table Look-up and Interpolation

E2. Curve Fitting (See also G2)

E3. Smoothing (see also G2)

E4. Extreme Values;   Includes variational problems.
Minimizing or Maximizing
a Function

F. OPERATIONS ON MATRICES, VEC-   This category shall include all
TORS AND SIMULTANEOUS LINEAR   routines whose primary purpose
EQUATIONS   is the execution of matrix opera-
tions, such as addition, multipli-
cation, inversion and transposition.
Evaluation of the determinant and
finding the eigenvectors and eigen-
values of a matrix shall be
included, as well as solutions of
a system of simultaneous linear
equations.

FO. General

F1. Matrix Operations

F2. Eigenvalues and Eigenvectors

F3. Determinants

F4. Simultaneous Linear Equations

G. STATISTICAL ANALYSIS AND   This category is meant to cover
PROBABILITY   routines which are basically
statistical in nature, such as
finding means, standard deviations,
etc., and routines like curve
fitting where the routine is
programmed explicitly for statis-
tical purposes.

G0. General

G1. Data Reduction   Refers to the calculation of the
more common statistical parameters
such as mean, median, standard
deviation, etc.

G2. Correlation and Regress- Includes curve fitting which is
ion Analysis   explicitly for statistical purposes.
(See also E3, E3)

G3. Sequential Analysis

G4. Analysis of Variance

G5. Time Series

G6. Monte Carlo (See also H) Includes Random Number Generators.

H. OPERATIONS RESEARCH TECHNIQUES, Operations research will mean the
    SIMULATION AND MANAGEMENT   application of scientific methods,
    SCIENCE   techniques and tools to the solution
                                   of problems involving operations
                                   of a system so as to provide those
                                   in control of the system with
                                   optimum solutions to the problem.
                                   Linear programming is a method of
                                   solving problems with objectives
                                   and constraints stated in a linear
                                   manner. Simulation is the technique
                                   of describing the operational
                                   systems by means of mathematical
                                   models. Management science, and
                                   interrelated areas, will be included
                                   and will cover such areas as
                                   solutions to problems of resource
                                   allocation. It is visualized that
                                   this category (H) will cover such
                                   things as production and inventory
                                   scheduling, logistics, queuing
                                   theory, renewal theory, dynamic
                                   programming, game theory and heur-
                                   istic programming. Monte Carlo
                                   methods used in this type of problem
                                   will be included although Monte Carlo
                                   methods may also appear in the
                                   statistical category (G) depending
                                   upon their use.

H0. General

H1. Linear Programming   Finding the best solution from
                                 among all solutions of a system
                                 of linear inequalities.

H2. Non-Linear Programming   Solving constrained optimization
                                 problems except those where the
                                 objective function and the
                                 constraints are all linear.

H3. Transportation and   Transportation codes utilizing
     Network Codes   efficient solution algorithms.
                                 Network codes to find maximal
                                 flow through a system.

H4. Simulation Modeling   "Simulation Modeling" is intended
                                 to encompass model components,
                                 general simulation programs, and
                                 simulation languages. For example,

I-24

GPSS or SIMSCRIPT would be categorized "Simulation Modeling." This category covers the tools and technology of simulation.

H5. Simulation Models

"Simulation Models" is intended to identify completed models of specific systems, however highly parameterized; e.g., a refinery model or computer job shop model would be categorized "Simulation Models." This category covers the completed products of model-making technology as, operating models of particular object systems.

H6. Critical Path Programs

H7. (Reserved. Please do not use.)

H8. Auxiliary Programs

Special purpose utility programs or subprograms designed especially to service programs in the above categories.

H9. Composite

Programs performing combinations of the above functions.

I. INPUT (See also A4)

Input shall contain those routines which transfer information from an external medium to an internal storage or operation area, e.g., from cards to an on-line magnetic tape. Input shall also include the programs which can perform either input or input and output operations. The categories shall denote the form of the information to be input. In classifying a program, the operation of input shall take precedence over the operation of conversion or internal information transfer (i.e., I rather than M2 or K4).

I0. General

I1. Binary

Pertains to program input or data input in the binary mode.

I2. Octal or Hexadecimal   Pertains to program input or data input in octal or hexadecimal mode.

I3. Decimal   Pertains to program input or numeric data input in the decimal mode.

I4. Character (Hollerith)   Pertains to program input or data input in the Character or Hollerith mode.

I5. (Reserved. Please do not use.)

I9. Composite   A combination of any of the above, which is not primarily one of the above, such as a general purpose input program.

J. OUTPUT (See also A4 and K)   Output shall contain those routines which transfer information from internal storage to an external medium. In classifying a program, output shall take precedence in general over conversion. The final form of the data relative to the program shall determine the category in which the program is classified.

J0. General

J1. Binary   Pertains to program output or data output in the binary mode.

J2. Octal or Hexadecimal   Pertains to program output or data output in the octal or hexadecimal mode.

J3. Decimal   Pertains to program output or data output in the decimal mode.

J4. Character (Hollerith)   Pertains to program output or data output in the character mode.

13-8

J5. Plotting

Refers to routines for producing plotted output, either via printer or via CRT, or other special plotting devices. Routines for using plotting devices to simulate printing are also included. Also includes routines for producing input for off-line plotters.

J6. (Reserved. Please do not use.)

J7. Analog (See also T6)

Refers to routines which output information to a digital-to-analog converter, other than that associated directly with (on- or off-line) plotting equipment, which will carry a J5 classification.

J9. Composite

A combination of any of the above, which is not primarily one of the above, such as a general purpose output program.

K. INFORMATION TRANSFER

The category shall contain routines whose primary purpose is the transfer of information from one internal medium to another internal medium or from one external medium to another external medium. If the transfer is external to internal or vice versa the categories I or J should be used.

K0. General

K1. External-to-External

Pertains to the transfer of information from any external device to any other external device. This would be tape-to-tape, drum-to-drum, disk-to-tape, etc.

K2. Internal-to-Internal

Pertains to the transfer of information internally. This is the same as relocation of information.

K3. (Reserved. Please do not use.)

13-9

K4. (Reserved. Please do
     not use.)

K5. Direct Data Devices   Computer-to-computer information
                              transfer, other than via the above
                              categories.

K9. Composite

L. EXECUTIVE ROUTINES

    L0. General

    L1. Assembly   A program system which assembles
                              a program for a particular problem
                              by assigning fixed addresses,
                              translating from mnemonic codes
                              to produce machine language state-
                              ments, possibly incorporating
                              library subroutines.

    L2. Compiling   Program systems which produce a
                              specific program for a particular
                              problem by translating from a
                              problem- or procedure-oriented
                              source language to produce a
                              machine language program. Inter-
                              pretive compilers should be
                              included.

    L3. Monitoring   Executive or monitor programs,
                              including monitor systems, to
                              control job sequencing, stacking,
                              log and accounting information,
                              peripheral equipment, etc., with
                              a minimum of operator intervention.

    L4. Preprocessing   Refers specifically to preprocessors
                              for programs within the "L" cate-
                              gory. A PERT (e.g.) preprocessor
                              would be cataloged H8 rather than
                              L4.

    L5. Disassembly and
         De-Relativizing

    L6. Relativizing

13-10

L7. Computer Language to Computer Language Translators

This refers to translation from one artificial language designed for computing and data processing purposes to another such language, e.g., FORTRAN to PL/1. Not to be used for translation of natural languages such as English or Russian.

M. DATA HANDLING

Data Handling shall refer to routines whose primary purpose basically is a mechanical type of handling of information such as sorting, conversion collating and merging, system tape editing (updating, deleting, adding to, etc.).

M0. General

M1. Conversion and/or Scaling

Pertains to any conversion and scaling routine (packed or unpacked, single or multiple precision) such as card image to BCD, BCD to card image, binary to BCD, BCD to binary, fixed to floating, etc. The primary function of programs in this category must be conversion or scaling, not input-output.

M2. Sorting

Combined sort/merge routines shall be included here.

M3. Merging

M4. Character Manipulation

M5. Searching, Seeking, Locating

To be used for utility search subroutines. Not to be used for applications of retrieving information records by examining contents, which is the province of Code S, Information Retrieval.

M6. Report Generators (See also T5)

M9. Composite

N. DEBUGGING   Debugging routines are those whose
                                     primary purpose is to provide
                                     information to a programmer to
                                     enable him to detect errors in
                                     software or programs. This will
                                     include dumps whose purpose is
                                     for debugging, as opposed to
                                     output, and search for debugging,
                                     as opposed to table look-up.

   N0. General

   N1. Tracing: Trapping,
        Snapshotting

   N2. Dumping (Post mortem)   Core, tape, drum, disk, console
                                     printouts (on- or off-line).

   N3. Memory Verification
        and Searching

   N4. Break Point Printing

O. SIMULATION/EMULATION OF COMPUTERS AND DATA PROCESSING:
    INTERPRETERS

   00. General

   01. Off-line Equipment   Any program which simulates
                                     off-line equipment.

   02. (Reserved. Please do
        not use.)

   03. Computers   Pertains to programs which
                                     simulate or emulate one computer
                                     on another.

   04. Pseudo-computers   Simulation of theoretical or
                                     pseudo-computers.

   05. Software simulation of   Includes such programs as simulating
        one peripheral device on   tape on disk, simulating card reader
        another   on tape, etc.

   09. Composite

P.   DIAGNOSTICS   Diagnostics programs shall be ones which check for malfunctioning of the computer or its components.

     P0.   General

Q.   SERVICE OR HOUSEKEEPING: PROGRAMMING AIDS   These are routines of a utilitarian nature which perform a service for the programmer or operator. This shall include routines to find check sums, clear memory and/or compare tapes, etc., and mechanical operations involving magnetic tapes.

     Q0.   General

     Q1.   Clear/Reset Programs

     Q2.   Check Sum Accumulation and Correction

     Q3.   Rewind, Tape Mark, etc. Programs

     Q4.   Internal Housekeeping; Save, Restore, etc.

     Q5.   (Reserved. Please do not use.)

     Q6.   Program Documentation: Flow Charting, Document Standardization, etc.

R.   NON-NUMERICAL MATHEMATICS   Logical functions, logical operations, logical calculi and algebras, symbol manipulation and manipulation of non-numeric quantities.

     R0.   General

     R1.   Formal Logic

     R2.   Symbol Manipulation   Includes combinational generators, permutations, etc.

     R3.   List and String Processing

S. INFORMATION RETRIEVAL   Programs, or systems, for cataloging
                                       data, all related to one field of
                                       interest, so it can be displayed,
                                       with accuracy and speed, on an
                                       output unit at any time it is
                                       needed. May or may not be real-
                                       time. Examples are classification,
                                       indexing and machine searching
                                       systems.

    S0. General

T. APPLICATIONS AND   Programs or subroutines which
    APPLICATION-ORIENTED PROGRAMS   solve specific problems in fields
                                       other than computers.

    T0. General

    T1. Physics (including
         nuclear)

    T2. Chemistry

    T3. Other Physical Sciences   Programs pertaining to physical
                                       sciences not listed above.
                                       (Geology, Astronomy, etc.)

    T4. Engineering

    T5. Business Data Processing
         (See also H and M6)

    T6. Manufacturing (non-data)
         Processing and Process
         control (See also J7)

    T7. Mathematics and Applied
         Mathematics

    T8. Social and Behavioral
         Sciences and Psychology

    T9. Biological Sciences

U. LINGUISTICS AND LANGUAGES   Programs which work with natural
                                 languages, usually translating
                                 from one language to another
                                 language. The rules of these
                                 languages reflect and describe
                                 current usage rather than prescribed
                                 usage. Example would be translating
                                 English to French.

    U0. General

V. GENERAL PURPOSE UTILITY SUBROUTINES

    V0. General

    V1. (Reserved. Please do
         not use.)

    V2. (Reserved. Please do
         not use.)

X. DATA REDUCTION   Many laboratory or field tests
                                 and experiments automatically
                                 record data either at the site
                                 or by way of telemetry. Programs
                                 in this category will accept such
                                 digital data and perform the
                                 necessary functions of decommuta-
                                 tion, scaling, calibrating,
                                 evaluating and test analysis.
                                 Some of the programs, especially
                                 X4, X5 and X6, might be predom-
                                 inantly of type D, E, F, G, and
                                 could possibly be found under
                                 those categories. The programs
                                 are either for post-processing
                                 or they may be on-line operation
                                 in real time.

    X0. General

    X1. Reformatting,   Program separates the variables
         Decommutation,   and often converts them to com-
         Error Diagnosis   puter words or higher level
                                 language variables. Missing or
                                 erroneous data are identified.
                                 Output consists of ordered data
                                 and editing information.

X2. Editing   Output from category X1 is used. Bad data are cast out, missing values inserted, wrong values corrected. Process is either automatic or by parameter cards, or both. Output is called clean **raw** data.

X3. Calibration   Data are scaled linearly, then calibrated to obtain function values in physical units. Output is called "clean calibrated data."

X4. Evaluation   All necessary computation on the data is performed to present them in a form suitable for engineering or scientific evaluation.

X5. Analysis   All computations necessary to analyze the outcome of the test or experiment. Also referred to as Time Series analysis.

X6. Simulation   Programs which generate artificial data to be used as the theoretical test model or to be used for checkout of programs.

Y. **USER ASSIGNED**   This category is for the use of individual installations who have groups of routines which do not fit well under any of the above categories. Routines from this category must be re-classified on an individual basis when and if they are submitted to the User Program Library.

Z. **ALL OTHERS**   This category contains all routines for which no primary class has not been designated. Routines which are covered by a primary class but which are not adequately described in a sub-class are assigned the applicable primary classification with a sub-class designation of zero.

1.0 IDENTIFICATION

1.1 Title

Real Zeros of a Single-Valued Function

1.2 Identification Name

DO-NRL-FCNZERO

1.3 Classification Code

DO - Operations on Functions and Solutions of Differential
Equations, General

1.4 RCC Identification Number

DO001ROO

1.5 Entry Points

FCNZERØ

1.6 Programming Language

Language: FORTRAN
Routine Type: Subroutine
Operating System: DRUM SCOPE 2.0

1.7 Computer and Configuration

CDC 3800

1.8 Contributor or Programmer

Janet P. Mason, Code 7813, Research Computation Center,
               Mathematics and Information Sciences Division

Harold L. Toothman, Code 4319H, Naval Analysis Staff,
               Office of Director of Research

1.9 Contributing Organization

NRL - Naval Research Laboratory - Washington, D.C. 20390

1.10 Program Availability

   1.10.1 Submittal: Program write-up, Fortran source deck, source
                        listing

   1.10.2 On File: RCC Program Library

1.11 Verification

   The following two problems were used as test cases:

   (1) Seventy zeros of the function $\sin(1/X)$. See Section 7.0.

   (2) $$\frac{F(\theta/T_2)}{F(\theta/T_1)} - \frac{(R_{02} - R_{r2})T_1}{(R_{01} - R_{r1})T_2} = 0$$

   where $0 \leq \theta \leq 6160$ and T_1, T_2, R_{01}, R_{02}, R_{r1}, and R_{r2}
   are known.

1.12 Date

   1 August 1968

2.0 PURPOSE

2.1 Description of the Routine

   To find the real zeros of a single-valued function of one real
   variable by a modified Method of False Position.

   Find within a specified tolerance δ those points x_i along a given
   closed interval $[a,b]$ for which the functional value $f(x_i)$
   satisfies: $|f(x_i)| < \delta$. Each point and the corresponding
   functional value are stored in a table:

   $$x_0, \ f(x_0), \ x_1, \ f(x_1), \ldots, \ x_n, \ f(x_n)$$

   where: $a \leq x_0 < x_n \leq b$

2.2 Problem Background

   See the authors.

3.0 USAGE

3.1 Calling Sequence or Operational Procedure

   FCNZERØ (FØX, XØWER, UPPER, TØLE, MAXRTS, NUMB, ANS)

3.2 Arguments, Parameters, and/or Initial Conditions

FØX is the name of a FORTRAN function, to be written by the user,
which evaluates the function whose zeros are to be calculated.
FØX must be declared external.

XØWER is the lower bound of the given interval.

UPPER is the upper bound of the given interval.

TØLE is the tolerance set upon both x_i and $f(x_i)$.

MAXRTS is the number of zeros (less than 513) the user expects to
find in the given interval. The user can make MAXRTS equal
to 0, thereby leaving the determination of the number of zeros
(e.g., roots) entirely up to the routine.

NUMB will contain the number of zeros actually found. NUMB may or
may not equal MAXRTS.

ANS is the name of the table of answers. The user should
dimension ANS(2050) or ANS(4 x MAXRTS).

3.3 Space Required (Decimal and Octal)

3.3.1 Unique Storage: 1020 octal (528 decimal) locations
exclusive of computer system library functions

3.3.2 Common Blocks:

CØMMØN/A/ INPUT,ØUTPUT,LIM,VAL,Z0,Z1,FZ0,FZ1

```
INPUT  ⎫
OUTPUT ⎪         LIM - integer
VAL    ⎪
Z0     ⎬ - real
Z1     ⎪
FZ0    ⎪
FZ1    ⎭
```

length: 10 octal (8 dec) locations

3.3.3 Temporary Storage: none

3.4 Messages and Instructions to the Operator

3.5 Error Returns, Messages, and Codes

3.6 Informative Messages to the User

Ordinarily there is no printout. If, however, the limit (100) on
the number of iterations in the False Position method has been

reached, or if the two points x_s and x_t bounding the true zero satisfy:

$$\left|(x_s - x_t)/x_s\right| < 0.0000000001,$$

then the following information is printed:

number of iterations, x_s, $f(x_s)$, x_t, $f(x_t)$.

The last point calculated is taken to be the zero (e.g., root) and is stored, as is its functional value.

3.7 Input

N. data input.

3.8 Output

Ordinarily, none; see 3.6.

3.9 Format

not applicable

3.10 External Routines and Symbols

FCNZERO - Q8QDICT.   (system library)
          CØNVERGE   (deck)

CØNVERGE - THEND.
           Q8QDICT. $\left.\right\}$ (system library)
           STH.
           QNSINGL.

3.11 Timing

18 seconds for sample program (e.g. to find at least 70 zeros of the function $\sin\left(1/x\right)$ $.00001 \leq x \leq 1.0$); see 5.0

3.12 Accuracy - - -

3.13 Cautions to Users

If the programmer wishes to find the first n zeros in a given interval he should let MAXRTS = 0 (see 3.2) in his subroutine call and pick out the first n zeros from the table of answers. If MAXRTS were to be made equal to n (and n is less than the actual number of zeros in the given interval), it is quite likely that the zeros found would not be the first n zeros in the interval.

3.14 Program Deck Structure

JØB card
FIN card
main program deck (contains call to FCNZERØ)
function deck (a single-valued function of one real variable) ⎱(See Sec-
⎰ tion 7.0)
subroutine FCNZERØ ⎱ DC-NRL-FCNZERØ (See Section 5.0)
subroutine CØNVERGE ⎰
SCØPE card
LØAD card
RUN card

3.15 References - Literature - Appendices

Introduction to Numerical Analysis by F. B. Hildebrand

4.0 METHOD OR ALGORITHM

After the number and approximate location of zeros are determined by
successive sectioning, the function employs a modified Method of False
Position to determine the exact location, within the given tolerance,
of these zeros. False Position (iterated linear inverse interpolation)
although known to converge, may do so very slowly. To hasten conver-
rence, the usual False Position method has been modified; the stepsize
in interpolation is adjusted whenever it becomes relatively small and
the last two successive approximations of the zero have the same sign.
Convergence is satisfied when either

$$(1) \quad |f(x_i)| < \delta \cap |x_t - x_s| < \delta ,$$

where $f(x_s) < 0$ and $f(x_t) > 0$, or

$$(2) \quad f(x_i) = 0.$$

5.0 FLOWCHART AND/OR SOURCE LANGUAGE LISTING

```
      SUBROUTINE FCN2ERO(LC,LB,UB,TL,NR,SC,LS )                        10
C     IDENT NUMBER - DC001R00                                          11
C     TITLE - REAL ZEROS OF A SINGLE-VALUED F NCTION                   12
C     IDENT NAME - DO4NRL-FCN7ERO                                      13
C     LANGUAGE - FORTRAN                                               14
C     COMPUTER - CDC-3600                                              15
C     CONTRIBUTOR - JANET P. MASON, CODE 7813. RESEARCH COMPUTATION    16
C                             CENTER, M S DIVISION                     17
C                   HAROLD L. TOOTHMAN, CODE 4319H, NAVAL ANALYSIS     18
C                                STAFF, OFFICE OF DIRECTOR OF          19
C                                RESEA CH                              20
C     ORGANIZATION - NRL - NAVAL RESEARCH LAB RATORY - WASHINGTON, D.C.21
C                                         20390                        22
C     DATE - 1 AUGUST 1968                                             23
C     PURPOSE - TO FIND THE REAL ZEROS OF A SINGLE-VALUED FUNCTION OF  24
C     ONE REAL VARIABLE BY A MODIFIED METHOD F FALSE POSITION          25
      DIMENSION BND(4), LST(2050)                                      26
      TYPE REAL LBOUT,INPUT,LB,LST,K4,LC                               30
      TYPE INTEGER TW,FSC,R,S,T,V,SC                                   40
      COMMON/A/INPUT,OUTPUT,LIM,VAL,Z0,Z1,FZ0.FZ1                      50
      LIM=1000                                                         60
      DELTAX=0.0                                                       70
      VAL=0.02                                                         80
      K4=0.367879441                                                   90
      TW=FSC=SC=R=V=0                                                  100
      DX=DELTAX                                                        110
      IF(DX.EQ.0)1,2                                                   120
    1 DX=UB-LB                                                         130
    2 IF(NR.EQ.0)3,4                                                   140
    3 NR=513                                                           150
    4 DUMMY=INPUT=LB                                                   160
      LBOUT=LC(DUMMY)                                                  170
    5 SC=R=0                                                           180
      BND(1)=LB                                                        190
      BND(2)=LBOUT                                                     200
    6 IF(BND(1).EQ.UB)7,8                                              210
    7 IF(ABS(BND(2)).LT.TL)9,10                                        220
    9 LST(R+1)=BND(1)                                                  230
      LST(R+2)=BND(2)                                                  240
      SC=SC+1                                                          250
   10 GO TO 20                                                         260
    8 INPUT=BND(1)+DX                                                  270
      IF(INPUT.GT.UB)11,12                                             280
   11 INPUT=UB                                                         290
   12 BND(3)=DUMMY=INPUT                                               300
      BND(4)=LC(DUMMY)                                                 310
      IF(BND(2).EQ.0)17,13                                            320
   13 IF(BND(2).LT.0)14,15                                             330
   14 IF(BND(4).GT.0)17,16                                             340
   15 IF(BND(4).LT.0)17,16                                             350
   16 GO TO 19                                                         360
   17 DO 18 I=1,4                                                      370
      LST(R+I)=BND(I)                                                  380
   18 CONTINUE                                                         390
```

```
        R=R+4
        SC=SC+1
        IF(SC.GE.513)25,19                                          400
     19 BND(1)=BND(3)                                                410
        BND(2)=BND(4)                                                420
        GO TO 6                                                      430
     20 IF(SC.EQ.FSC)21,22                                           440
     21 TW=TW+1                                                      450
        GO TO 23                                                     460
     22 TW=1                                                         470
     23 IF(SC.LT.AR.AND.TW.LT.4)24,25                                480
     24 DX=K4*DX                                                     490
        FSC=SC                                                       500
        GO TO 5                                                      510
     25 T=(SC-1)*4+1                                                 520
        DO 32 S=1,T,4                                                530
        IF(LST(S+1).EQ.0)26,27                                       540
     26 LST(V+1)=LST(S)                                              550
        LST(V+2)=LST(S+1)                                            560
        GO TO 31                                                     570
     27 IF(LST(S+1).LT.0)28,29                                       580
     28 Z0=LST(S)$    FZ0=LST(S+1)                                   590
        Z1=LST(S+2)$  FZ1=LST(S+3)                                   600
        GO TO 30                                                     610
     29 Z1=LST(S)$    FZ1=LST(S+1)                                   620
        Z0=LST(S+2)$  FZ0=LST(S+3)                                   630
     30 CALL CONVERGE(LC,TL)                                         640
        LST(V+1)=INPUT                                               650
        LST(V+2)=CUTPUT                                              660
     31 V=V+2                                                        670
     32 CONTINUE                                                     680
     33 END                                                          690
                                                                     700
                                                                     710
```

```
      SUBROUTINE CONVERGE(LCN,TLN)                                800
      TYPE REAL K3,INPUT,LCN                                      810
      COMMON/A/INPUT,OUTPUT,LIM, AL,Z0,Z1,FZ0,FZ1                 820
      FACTOR=1.0                                                  830
      K2=0                                                        840
      FOROUT=K3*1.0E100                                           850
   34 DUMMY=INPUT=(FZ1*Z0-FZ0*Z1)/(FZ1-FZ0)                       860
      OUTPUT=LCN(DUMMY)                                           870
      K2=K2+1                                                     880
      IF(ABS(OUTPUT).LT.TLN.AND.ABS(Z1-Z0).LT.TLN)50,35          890
   35 RATIO=ABS((FORIN-INPUT)/(Z1-Z0))                            900
      IF(RATIO.LT.VAL.AND.FOROUT*OUTPUT.GT.0)36,42                910
   36 FACTOR=FACTOR*10.0                                          920
   37 IF(RATIO*(FACTO-+1.0).GE.1.0)38,39                          930
   38 FACTOR=FACTOR/2.0                                           940
      GO TO 37                                                    950
   39 SPEC=INPUT                                                  960
      ALSO=OUTPUT                                                 970
      DUMMY=INPUT=INPUT-FACTOR*(FORIN-INPUT)                      980
      OUTPUT=LCN(DUMMY)                                           990
      K2 = K2+1                                                  1000
      IF(ALSO.LT.0)40,41                                         1010
   40 FORIN=Z0=SPEC                                              1020
      FOROUT=FZ0=ALSO                                            1030
      GO TO 42                                                   1040
   41 FORIN=Z1=SPEC                                              1050
      FOROUT=FZ1=ALSO                                            1060
   42 IF(OUTPUT.EQ.0)50,43                                       1070
   43 IF(OUTPUT.LT.0)44,45                                       1080
   44 FORIN=Z0=INPUT                                             1090
      FOROUT=FZ0=OUTPUT                                          1100
      GO TO 46                                                   1110
   45 FORIN=Z1=INPUT                                             1120
      FOROUT=FZ1=OUTPUT                                          1130
   46 IF(ABS(OUTPUT).LT.TLN.AND.ABS(Z1-Z0).LT.TLN)50,47          1140
   47 IF(K2.GT.LIM.OR.ABS((Z0-Z1)/Z0).LT.0.0000000001)48,34      1150
   48 PRINT 49,K2,Z0,FZ0,Z1,FZ1                                  1160
   49 FORMAT(1X,I4,3X,E17.10,3X,E17.10,3X,E17.10,3X,E17.10)      1170
   50 END                                                        1180
```

6.0 COMPARISON

No comparisons have been made with any other programs.

7.0 TEST METHOD AND RESULTS

The main program (TRIAL) and function evaluator (FOX) used to evaluate the seventy zeros of the function $\sin(1/X)$, along with the results as printed out by the main program, follow this page.

MAIN PROGRAM
(CALLS FCNZERO)

```
      PROGRAM TRIAL                                              001
      EXTERNAL FOX                                               S001
      DIMENSION ANS(2050)                                        T001
      NT=0                                                       U001
   4  FORMAT (1H1)                                               V001
      PI=3.1415926536                                            002
      MAXRTS=70                                                  003
      TOLE=5.0E-4                                                004
      XOWER=1.0E-5                                               005
      UPPER=1.0                                                  006
      CALL FCNZERO(FOX,XOWER,UPPER,TOLE,MAXRTS,NUMB,ANS)         007
      NUMB=NUMB+NUMB                                             008
      PRINT 3                                                    009
   3  FORMAT(1H1,17X,*X*,15X,*SIN(1/X)*,8X,*(1/X`/PI*//)         S009
      DO 1 J=1,NUMB,2                                            011
      FINAN=ANS(J)                                               012
      REC=1.0/(FINAN*PI)                                         013
      FUN=ANS(J+1)                                               014
      PRINT 2, FINAN,FUN,REC                                     015
   2  FORMAT(9X,E17.10,3X,E17.10,3X,F9.3)                        016
      NT=NT+1                                                    017
      IF(NT.GE.56)5,1                                            018
   5  PRINT 4                                                    019
      NT=0                                                       020
   1  CONTINUE                                                   021
      END                                                        022
```

FUNCTION EVALUATOR
(EVALUATES SIN 1/X)

```
      FUNCTION FOX(X)                                            01
      X=1.0/X                                                    02
      FOX=SIN(X)                                                 03
      END                                                        04
```

X	SIN(1/X)	(1/X)/FI
1.3340732483-004	2.1490885245-004	2386.000
5.0127540873-004	-1.0243733414-005	635.000
5.0365487938-004	1.5077646821-005	632.000
7.0893061138-004	-1.8468816311-004	449.000
9.4174529791-004	-8.1762761255-005	338.000
1.1090531983-003	-3.9168663260-004	287.000
1.1449595878-003	2.1455052774-006	278.000
1.2482740621-003	-8.3513441496-007	255.000
1.4338280848-003	1.1254008835-004	222.000
1.9770800348-003	-9.0987305157-007	161.000
2.0649473425-003	-6.9291007260-006	154.000
2.1080124913-003	1.6475331383-008	151.000
2.2736420442-003	-9.5486197339-009	140.000
2.3578510657-003	1.0244172534-005	135.000
2.8420519896-003	7.0021836961-005	112.000
3.0903892490-003	2.0988605786-004	103.000
3.2480574804-003	2.4481041811-004	98.000
3.4979106355-003	-1.6478938050-005	91.000
3.6171580596-003	-2.0030393731-005	88.000
3.8305088762-003	4.4779409874-007	83.000
4.0808961100-003	-8.0076595789-006	78.000
4.8970751410-003	3.7483079580-007	65.000
5.1340286638-003	6.9748522890-005	62.000
5.3950828603-003	1.4991408701-006	59.000
5.4880962515-003	1.0738559629-004	58.000
5.7874524493-003	-7.9983146861-007	55.000
5.8946275903-003	-1.9697789125-006	54.000
6.0058470626-003	4.2551280952-006	53.000
6.1213439390-003	6.9400994107-007	52.000
6.2413703814-003	1.6471629518-006	51.000
6.3661977443-003	-5.0897035036-007	50.000
6.4961175957-003	-5.9965619547-005	49.000
6.6314559734-003	-2.5541777943-007	48.000
6.7725439314-003	-1.4908969697-004	47.000
6.9197797743-003	7.5194693636-006	46.000
7.0735598010-003	1.3659880128-004	45.000
7.2343156302-003	-6.7215552774-007	44.000
7.4025717666-003	2.9698348773-004	43.000
7.5788092086-003	-4.1691094161-005	42.000
7.7636552472-003	-8.5165956989-006	41.000
7.9577476522-003	-7.8577457482-006	40.000
8.1617534015-003	2.1738060983-005	39.000
8.3765758420-003	1.5711702872-006	38.000
8.6029699114-003	1.9633700360-007	37.000
8.8419412966-003	-1.7573149164-007	36.000
9.0945681944-003	2.1659320960-007	35.000
9.3620554812-003	-5.9985557916-008	34.000
9.6457531932-003	-1.0031915735-005	33.000
9.9471435430-003	4.0630464925-004	32.000
1.0268057192-002	-3.4648868851-005	31.000
1.0610335341-002	-5.1537443863-005	30.000
1.0976211530-002	7.1035496832-005	29.000
1.1368210386-002	-1.2774274897-006	28.000
1.1789239332-002	-1.1304809571-004	27.000
1.2242667491-002	2.9267485022-006	26.000
1.2732395582-002	8.3190645312-007	25.000

A-11

1.3262904913-002	3.9657666696-005	24.000
1.3839560428-002	4.2978187350-007	23.000
1.4468631136-002	2.5995541364-007	22.000
1.5157615567-002	8.4384544773-006	21.000
1.5915494725-002	-1.6415459126-006	20.000
1.6753044592-002	-3.8234935142-004	19.000
1.7683882711-002	-4.6319002506-007	18.000
1.8724127098-002	4.6053147162-005	17.000
1.9894368018-002	-3.3347425272-007	16.000
2.1220609391-002	-1.1034033355-004	15.000
2.2736614539-002	-3.7546624037-004	14.000
2.4485578572-002	3.3811357182-004	13.000
2.6525837247-002	-1.9042347792-005	12.000
2.8937409685-002	1.7591417583-004	11.000
3.1831022059-002	-3.3004473891-005	10.000
3.5367871674-002	8.5173788376-005	9.000
3.9788830071-002	-5.9563698214-005	8.000
4.5472524489-002	4.0432819156-005	7.000
5.3051686926-002	-1.4648542508-005	6.000
6.3662016117-002	9.5932071584-006	5.000
7.9577506856-002	-5.5760636481-006	4.000
1.0610332581-001	2.7021185983-006	3.000
1.5915496642-001	-9.2097186435-007	2.000
3.1830580482-001	8.5221142264-008	1.000

8.0 REMARKS

   None

PROGRAM BRIEF

TITLE. Real Zeros of a Single-Valued Function

IDENTIFICATION NAME. DO-NRL-FCNZERO

PROGRAMMING LANGUAGE. 3600 FORTRAN

COMPUTER AND CONFIGURATION. CDC-3800

PROGRAMMER.   Janet P. Mason, Code 7813
              Harold L. Toothman, Code 4319H

NRL ORGANIZATION. Mathematics & Information Sciences Division
                   Research Computation Center
                   Code 7813

DATE. 15 January 1968

PURPOSE.   To find the real zeros of a single-valued function of one real
           variable by a modified Method of False Position.

PROGRAM STATUS. Complete

ADDENDUM C

CATALOG IDENTIFICATION : __DO-NRL-FCNZERO__   REVISION : _____

TITLE (DESCRIPTIVE) :   __Real Zeros of a Single-Valued Function__

_____

SOURCE LANGUAGE(S)
AND DIALECT(S) :   __3600 FORTRAN__

OPERATING SYSTEM
AND VERSION :   __DRUM SCOPE Version 2.0__

COMPUTER & CONFIGURATION : __CDC-3800__

                                    exclusive of computer
                __1020 octal locations (528 decimal) system library functions__

CONTRIBUTING ORGANIZATION : __NRL - Naval Research Laboratory, Washington, D.C. 20390__

                Janet P. Mason, Code 7813
SUBMITTOR/PROGRAMMER : __H. L. Toothman, Code 4319__ REVISOR : _____

DATE WRITTEN :   ORIGINAL __15 January '68__ REVISION _____

PROGRAM MATERIAL :   WRITE-UP PAGE COUNT : _____

                SEQUENCED SOURCE-CARD COUNT : __119___

                M/T RECORD COUNT : _____ P/T RECORD COUNT : _____

OTHER AVAILABLE DOCUMENTATION : _____

ORIGINAL / REVISED PROGRAM CATALOG ABSTRACT :

    A subroutine to find the real zeros of a single-valued function of one
real variable by a modified Method of False Position.
    Entry point: FCNZERO

NATURE OF REVISION (OR ADDITIONAL INFORMATION) :

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Naval Research Laboratory	UNCLASSIFIED
Washington, D.C. 20390	2b. GROUP

3 REPORT TITLE

RESEARCH COMPUTATION CENTER PROGRAM PUBLICATION GUIDE

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)
A final report on one phase of the problem; work is continuing.

5 AUTHOR(S) (First name, middle initial, last name)

Doris E. Gossett, Janet P. Mason, and Alan B. Bligh

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
February 1969	62	
8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)	
NRL Problem B02-03		
b. PROJECT NO	NRL Memorandum Report 1946	
RR 003-09-41-5101		
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.		

10 DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
	Department of the Navy (Office of Naval Research), Washington, D.C. 20360

13 ABSTRACT

A guide has been developed that contains the Research Computation Center standards for documenting, revising, classifying, submitting, identifying, obtaining, and evaluating computer programs at NRL. It is anticipated that the use of such standards will promote efficient organization of an RCC Program Library and will provide for maximum effective usage of this program library throughout NRL.

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Computer programs						
Format for reports						
Report standards						
Publication guide						

DD ,FORM 1473 (BACK)
1 NOV 65
(PAGE 2)