✓
SAND77-0618
Unlimited Release

MASTER

3151

QMESH/RENUM/QPLOT Mainframe and Plotter Conversion--CDC6600 to IBM360/370 and SC4020 to CALCOMP

A. Fred Schkade

Prepared by Sandia Laboratories, Albuquerque, New Mexico 87115
and Livermore, California 94550 for the United States Energy Research
and Development Administration under Contract AT(29-1)-789

Printed September 1977

Sandia Laboratories

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Issued by Sandia Laboratories, operated for the United States Energy Research & Development Administration by Sandia Corporation.

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research & Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America

Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Price: Printed Copy \$4.50 ; Microfiche \$3.00

SAND77-0618
Unlimited Release
Printed September 1977

**QMESH/RENUM/QPLOT MAINFRAME AND PLOTTER CONVERSION--
CDC 6600 TO IBM360/370 AND SC4020 TO CALCOMP**

A. Fred Schkade
Applied Mechanics I Division 1281
Sandia Laboratories
Albuquerque, NM 87115

NOTICE
This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

ABSTRACT

This is a report of a code conversion project to modify an existing set of FORTRAN computer programs to run on a different mainframe and to drive a different plotter. The mainframe conversion was from a CDC6600 to an IBM360; the plotter conversion was from an SC4020 (driven by calls to SCORS library subroutines) to a Calcomp. The changes made because of differences between the mainframes' FORTRAN compilers were relatively minor. IBM ASSEMBLER source decks were acquired to replace all of the CDC special features (such as ENCODE/DECODE) which have no direct IBM equivalents. The plotter call conversion was more difficult, but was still relatively straightforward. By far the most difficult part of the conversion effort involved converting the file-handling protocol from default-tolerant SCOPE on the CDC6600 to explicit JCL (Job Control Language) on the IBM360; listings of all the IBM deck setups used are included in the Appendix. The programs converted were QMESH, RENUM, QPLOT, RENUM8, and QPLOT8, which together constitute a general-purpose quadrilateral-element mesh generation preprocessor to a variety of finite-element stress analysis programs.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ACKNOWLEDGEMENTS

The author wishes to acknowledge the assistance and advice contributed by Harry Edwards of Argonne Code Center, and by Lester Petrie, John Carpenter, and Steve Margle of Oak Ridge National Laboratory.

Appreciation is also expressed to Bob Hagen and Ed Orner of the ADP Operations Branch, Management Information Systems Division, Albuquerque Operations Office, Energy Research and Development Administration, for making available the use of their IBM mainframe and Calcomp plotter for testing the converted programs, and to the following members of their staff who provided much-needed assistance and consultation: Bill Chaney, Dee Dee Ranyan, Sherry Schluter, Ron Sikorski, and Carol Valdez.

This work was sponsored by the Fuel Cycle Research Branch, Division of Safeguards, Fuel Cycle, and Environmental Research, Office of Nuclear Regulatory Research, U.S. Nuclear Regulatory Commission.

CONTENTS

	<u>Page</u>
I. Introduction	7
II. Background	8
III. Preliminary Steps	8
IV. Conversion Procedure	9
V. Software Modification Details	11
VI. Test Results and Example Problem	19
VII. Conclusion	29
References	30
APPENDIX A -- IBM Deck Setups Used on the ERDA/ALO IBM360/50 During the Program Conversion Effort	31
APPENDIX B -- QMESH and RENUM Input Data Decks for the Example Shipping Cask Problem	61

FIGURES

<u>Figure</u>	<u>Page</u>
1 QMESH Plot of Shipping Cask Mesh	21
2 Annotated QMESH Plot of Shipping Cask Region 17	22
3 Annotated QMESH Plot of Shipping Cask Region 20	23
4 QPLOT Plot of Shipping Cask Section 1	25
5 QPLOT Plot of Shipping Cask Section 11	26
6 QPLOT Plot of Shipping Cask Section 12	27
7 QPLOT Map Showing Locations of Plotted Sections	28

TABLES

<u>Table</u>	<u>Page</u>
I Comparison of Compilation Time and Central Memory Requirements For the IBM and CDC Versions	20
II Comparison of Test Problem Run Times for the IBM and CDC Versions	20

QMESH/RENUM/QPLOT MAINFRAME AND PLOTTER CONVERSION--
CDC 6600 TO IBM360/370 AND SC4020 TO CALCOMP

I. Introduction

QMESH [1,2,3] is a general-purpose computer program for producing two-dimensional meshes of quadrilateral elements for finite element programs. It incorporates automatic techniques for mesh smoothing and restructuring--not previously available in a general-purpose mesh generation program--which make it much easier to achieve a regular, uniform grid network. Considerable effort was applied toward providing an easy, flexible data input method free from artificial restrictions. The output from QMESH consists of: (1) a raw data file containing the calculated nodal point coordinates and element connectivity information; (2) a line-printer plot of the boundary nodes of each "region" of the mesh, which is useful for early detection of input data errors; and (3) a plot file containing the complete nodal point and element layout.

A companion program, RENUM, processes the raw data file from QMESH and assigns nodal point numbers so as to minimize the bandwidth of the resulting system of equations and thereby reduce the computation time required for its solution. Since RENUM also performs certain data cleanup and formatting functions, it must always be used on the QMESH results. The output from RENUM is a data file which is ready to be used as input to a finite element program.

A second companion program, QPLOT, is available to plot the final grid with nodes and elements labeled--an almost indispensable aid to interpreting the finite element results. The only input required by QPLOT is the data file created by RENUM.

While RENUM and QPLOT process four-node quadrilateral elements with nodes at each corner, modified versions of these programs, called RENUM8 and QPLOT8, are available to process eight-node quadrilaterals which have additional mid-side nodes along each element edge. The same QMESH driver program is used for input to both RENUM and RENUM8.

The five programs described above were originally written for the CDC 6600 mainframe. The plot files generated by QMESH, QPLOT, and QPLOT8 were for the SC4020 optical printer. The objective of the work reported herein was to convert these programs to run on an IBM360 or 370 mainframe and to drive a Calcomp plotter.

II. Background

In the first quarter of 1975, Sandia Laboratories submitted a proposal [4] to the Nuclear Regulatory Commission (NRC) defining a project in which Sandia was to upgrade NRC's capability to perform structural analyses of shipping containers. The proposal was accepted by NRC [5] in December 1975. Among the tasks specified in the proposal was a project to provide NRC with several structural analysis computer codes and their associated pre- and postprocessors. An interim report [6], issued in April 1976, announced that QMESH had been chosen as the preprocessor because of its flexibility, solidity, and compatibility with a number of structural analysis codes.

As noted above, QMESH and its associated programs were written for the CDC6600 computer and SC4020 plotter. Because of existing budgetary and/or consultation agreements, however, NRC insisted that all codes provided to them by Sandia be converted to run on IBM360/370 mainframes. Specifically, NRC specified that all codes were to be installed and made operational on the IBM360/91 and its associated Calcomp plotters at Oak Ridge National Laboratory (ORNL), to which NRC had a communications link through an IBM 2922 Remote Work Station.

Before the conversion effort was begun, a visit to ORNL and NRC/Bethesda [7] was arranged in order to consult with the personnel who would be involved in implementing and using the programs and to observe and evaluate the target computer equipment. The IBM2922 terminal at Bethesda is a medium-speed nonintelligent terminal and does not have any remote plotting equipment--a serious drawback considering the enormous benefits of rapid plots to the structural analyst. Part of the planning for an upcoming NRC move to Silver Spring, however, includes the acquisition of a high-speed COPE intelligent remote terminal with an on-line plotter; observation of a similar terminal in operation at Bethesda indicates that it will be quite adequate to support the converted codes.

This report is a chronicle of the work performed to accomplish the program conversion. Hopefully, it will be useful as a guide to others who are faced with the task of performing similar conversions.

III. Preliminary Steps

The first step in the conversion effort involved acquiring and studying pertinent reference materials. These included publications describing the particular target computer equipment and operating procedures [8, 9], reference publications from IBM [10, 11] and CDC [12, 13, 14], manuals describing the features of Sandia's CDC6600 computer [15] and SC4020 plotter [16],

and a user's guide to "standard" Calcomp software [17]. Two publications [18, 19] outlining the differences between IBM and CDC Fortran were particularly helpful.

There are no IBM computers in the Sandia Laboratories computer complex. An informal arrangement was negotiated with personnel at the ERDA/ALO headquarters whereby their IBM360/50 computer and Calcomp plotter were made available for testing the converted programs.

The original CDC6600 versions of QMESH and RENUM (but not QPLOT, RENUM8, or QPLOT8) were released to Argonne Code Center in July 1974. During the summer of 1975, these two programs were converted by Argonne personnel to run on an IBM360 or 370 mainframe and to drive a Calcomp plotter. A tape of the Argonne versions was obtained, primarily for use as a reference to help identify the types of code modifications required; it was extremely useful in this respect. The Argonne versions, however, did not contain the error corrections and logic improvements of the latest Sandia versions; no attempt had been made to employ acceptable IBM substitutes for many useful CDC system subprograms; and the very important plotter call conversions were rather hastily, although adequately, done.

As alluded to above, the CDC6600 versions of these five programs make use of several library subprograms and special Fortran features which, for most IBM installations, are either substantially different (e.g., elapsed CPU time) or unavailable (e.g., ENCODE/DECODE). To simplify the program conversions and to avoid making existing documentation obsolete, it seemed desirable to utilize equivalent IBM subprograms and features on a one-to-one replacement basis wherever possible. An extensive selection of on-line library subprograms [9, ch. 20] is available on the Oak Ridge computing system; assembler decks for fourteen of these routines were requested for evaluation of their potential use by the converted programs.

IV. Conversion Procedure

Once the preliminary steps were completed, it seemed advisable to acquire some "hands on" experience with the unfamiliar IBM equipment and procedures. To accomplish this, the Argonne versions of QMESH and RENUM were installed, compiled, and tested on the IBM360/50 at ALO, and were made to execute correctly with only a few modifications to the plotter calls. The changes which had been made by Argonne subsequently served to guide the conversion of the latest Sandia versions.

Additional familiarization with the IBM Fortran compiler and Job Control Language (JCL) was acquired by individually testing each of the ORNL-supplied Assembler routines at ALO. The possibility that NRC might subsequently choose to move the converted programs from Oak Ridge to some other IBM mainframe was a very practical consideration, and if the ORNL routines also worked correctly at ALO it was felt that they would probably do so on most IBM mainframes.

Short Fortran drivers were written to perform the tests, and only two of the Assembler routines--neither of which was subsequently needed for the conversion--failed to work at ALO.

Some experimentation was performed to develop capabilities which were either not readily obtainable from the ORNL computer library or, if available, would have been too inefficient to use (e.g., a very small portion of a large, general purpose routine). These capabilities included bit shifting, handling of literal (Hollerith) arguments in subroutine calls, and character manipulation within literal strings. It was found that character (byte) manipulation can be performed in Fortran H on an IBM360 much more easily than was expected: simply by declaring the incoming literal string array to be "type" LOGICAL*1 in the called subroutines, each character in the string can be individually addressed and manipulated.

The decision was made to perform the bulk of the program conversion on Sandia's CDC6600 for a number of reasons: familiarity with the CDC equipment and operating system and unfamiliarity with the IBM counterparts--particularly the Job Control Language (JCL); the proximity and accessibility (nearby office) of a high-speed remote terminal to the CDC6600, whereas the IBM360 at ALO is approximately a mile away, is operated in a closed-shop environment (to non-ALO users), and during prime shift supports the INTERCOMM teleprocessing system which requires so much memory that the Fortran H compiler will not fit into core with it; Ron Jones, the author of the QMESH programs, was available at Sandia for consultation and assistance, if needed; and we did not wish to jeopardize our "gentlemen's agreement" with ALO's MIS division by degrading the throughput of their own jobs or by monopolizing the time of their staff members.

The latest Sandia versions of the five programs were placed on the disk-resident permanent file system in UPDATE "program library" format; UPDATE is CDC's source deck editing utility, and was used to apply the IBM and Calcomp modifications to the original decks under separate IDENTs so that the effects of each could be tested separately. The Fortran Extended (FTN) compiler was used to check the validity of the source code modifications; although the Fortran features unique to IBM generated CDC compiler diagnostics (as expected), the R = 3 listing option still supplied extensive cross-reference printouts of entry and exit points, variable names and references, I/O file names and references, external subroutine and function references, and statement number definitions and references, which were manually scanned to detect any modification errors or inconsistencies. The CDC loader and COLLECT processors were used to produce an extended core map listing which was manually scanned to verify that all references to SCORS plotting routines and special CDC in-line functions had been eliminated or replaced by their Calcomp or IBM counterparts. Finally, the Calcomp modifications were bench-tested using actual QMESH input by supplying surrogate subroutines which printed out the parameters transferred by each call to a Calcomp routine.

When the modified programs had been tested as thoroughly as was possible on the CDC6600, the final card images were written to a seven-track magnetic tape in External BCD format for transfer to the IBM360 at ALO. They were loaded into the IBM disk file system using the Librarian, a proprietary software product marketed by Applied Data Research, Inc. which provides source editing and file updating capabilities similar to CDC's UPDATE processor. One very powerful feature of the Librarian--not available from UPDATE--is the ability to replace every incidence of a particular character with another one. Using this feature, it was a simple matter to correct the characters that were erroneously translated from BCD into EBCDIC: & to +, % to (, < to), # to =, and ' to ; .

Once installed on the 360, the five programs were extensively tested using special test decks as well as actual structural analysis meshes, and the IBM output listings and Calcomp plots were compared with corresponding results from CDC6600 runs. Only two errors were discovered in the converted programs, one of which resulted from the decreased accuracy of IBM's 32-bit floating-point word.

Listings of all of the deck setups used on the ERDA/ALO IBM360/50 are contained in Appendix A.

V. Software Modification Details

At the beginning of the conversion effort, the UPDATE program library created for each of the five codes was catalogued into the permanent file system in the following format: each main program, subroutine, or function was assigned a "DECK name" the same as the program, subroutine, or function name; and the subroutines and functions were arranged in alphabetical order following the main program (which itself followed all COMDECKs) to simplify the task of locating each routine in the listings.

The code modifications were made under two different IDENTs: /IBM/ if the change were being made because of differences between CDC and IBM Fortran and/or system libraries, and /CAL/ if the change were being made because of differences between the SC4020 and Calcomp plotting packages. When the final card image file was created on the CDC6600 (the COMPILE file produced by UPDATE), the "8" option for 80-column card images was specified on the UPDATE card so that the correction IDENTs and their ascending sequence numbers would appear in columns 73 through 80 interspersed with the DECK names and original sequence numbers of the unmodified cards. All cards deactivated in the original versions were reinserted as comments with the characters "C-----" in columns 1 through 6, except that any nonblank characters in these columns on the original cards (e.g., statement numbers, continuation characters) were retained.

The types of changes made in converting the source code from the CDC 6600 to the IBM 360 (the /IBM/modifications) are summarized below.

1. PROGRAM card -- The card was deleted from each main program.
2. COMMENT card -- A "preface block" of comment cards was inserted in each main program and describes, in a general way, the modifications which were performed along with the calling sequences for the IBM Assembler and Calcomp subroutines which were used.
3. Hollerith (literal) variables -- 8- character real variables (REAL*8) were substituted for 10-character Hollerith variables wherever practical; literal arrays composed of 10-character words were converted to arrays of 4-character words (the default size) having a character capacity at least as large as the originals.
4. Implied DO loops in DATA declarations -- Implied loops of the form DATA (ARRAY (I, I = i_1, i_2) /---/ were converted to the form DATA ARRAY /---/, in which the dimensioned size of the array is automatically used.
5. REMARK calls -- Calls to the CDC system routine REMARK, which inserts the specified character string into the job's "dayfile", were either deleted or replaced by normal printouts.
6. FTNBIN calls -- Calls to the CDC system routine FTNBIN, which produces more efficient blocking of binary files, were deleted.
7. HOROLOG calls -- Calls to the system routine HOROLOG, which returns the current date and time along with the number of CPU seconds remaining (the latter was not used), were replaced with calls to IDAY and TIME.
8. Title/header printout -- The "latest revision" printout was changed, the IBM core requirement was substituted for the CDC value, and the time/date formats were changed from A10 to A8.
9. Error abort -- The intentional inducement of a MODE 2 arithmetic error (division by zero) to bypass subsequent control card processing upon detection of an unrecoverable error condition was replaced by a call to ABEND.
10. STOP statements -- STOP statements were inserted in all main programs, as required by the IBM compiler, and were used to replace CALL EXIT statements wherever they appeared; also, STOP 3333 statements were inserted following all calls to ABEND upon detection of any unrecoverable errors, so that the "user condition code" feature available on IBM mainframes might be used to bypass subsequent JCL steps in case the ABEND routine provided does not work correctly at a particular installation.

- 12
11. SECOND calls -- Calls to the system routine SECOND, which returns the current value of elapsed CPU time, were replaced with references to function ICLOCK in the following form:
CPUTIM = FLOAT (ICLOCK (IDUMMY))/100.0.
 12. Literals in FORMAT statements -- Asterisk-delimited character strings were replaced by their standard Hollerith forms (e. g., *ABC* was changed to 3HABC).
 13. ENCODE statements -- All ENCODE statements used to translate variables of all types into literal strings were replaced by calls to subroutine CORE followed by WRITE statements, as illustrated in the following example:

Before: ENCODE (nchar,nfmt,arrrto) arrfrm

After: CALL CORE (arrrto,nbytes)

WRITE (idummy,nfmt) arrfrm

where nbytes = number of bytes allocated to "arrrto",

and idummy = any logical unit number.

Used in this form, the call to CORE instructs the IBM system to ignore the unit number "idummy" and, instead, to write the variable list "arrrfrm" according to format number "nfmt" into the in-core buffer "arrrto"; following execution of the WRITE statement, "arrrto" contains the desired literal string.

14. End-of-file detection -- Use of the system function EOF to detect end-of-file following execution of a READ statement was replaced by using the "END=" parameter available with the IBM READ statement.
15. Left-justification of character strings--The original version of subroutine JLEFT, which used CDC-resident bit-shifting and masking functions, was replaced by one which utilizes a special capability of the IBM Fortran "H" compiler to individually examine and manipulate each character in a literal string by declaring the incoming array to be LOGICAL*1; in addition to left-justifying (in place) the incoming character string, the new JLEFT was expanded to return also the final length of the string.

16. DECODE statements -- All DECODE statements used to translate literal string into variable values were replaced by sequences of calls to subroutine CORE followed by READ and WRITE statements as illustrated in the following example:

```
Before:  DECODE (nchar,nfmt, arrfrm) arrto
After:   CALL CORE (ibuff,nbuff)
        WRITE (idummy, nnn) arrfrm
        nnn FORMAT (A-format corresponding to literal
                string contents)
        CALL CORE (ibuff,nbuff)
        READ (idummy, nfmt) arrto
where    ibuff is an in-core array to be used for the WRITE/
        READ sequence,
        nbuff is the size, in bytes, of ibuff,
        and idummy is any logical unit number
```

Used in this form, the first call to CORE directs the system to perform the subsequent WRITE to the in-core buffer, while the second call to CORE directs the system to perform the subsequent READ from the buffer; in both cases the unit number "idummy" is ignored.

17. Tolerance tests -- Because of the reduced accuracy of the IBM's 32-bit floating-point word, it was necessary to relax the accuracy tolerances in two subroutines in order for certain geometric tests to be performed correctly; since the original tolerance values seemed to be inappropriately tight, the use of double precision in the IBM versions was not felt to be necessary.
18. Added printouts -- To provide output uniformity, title/header and execution-time printouts similar to those produced by QMESH, RENUM, and RENUM8 were added to QPLOT and QPLOT8; the original versions of QPLOT and QPLOT8 produced no output listing at all unless a data error was detected.

The plotter called conversions required extensive modifications, owing to the great dissimilarity between the SC4020 and Calcomp driver software routines. By far the most significant difference is that the low-level SC4020 routines expect coordinate values to be passed in integer raster units (range 0 to 1023 in both X and Y), whereas the equivalent Calcomp routines expect real-valued coordinates in inches (acceptable range depending on plotter size). Similarly, the character sizes and spacing for the SC4020 are defined in rasters as opposed to inches for the Calcomp.

Since Calcomp plotters are available in a variety of sizes, it seemed advisable to set up the appropriate plotter size parameters, and character sizes and spacing, as variables in DATA declarations and to make all axis, vector, and label placement calculations based on the values of these variables. A number of new variable names were introduced into the codes with the intent of making the resulting logic easier to understand; for example, most of the variables representing physical plotter dimensions and coordinates (in inches) begin with the letter "P".

In order to assist in any future modifications of the plotting format, the following list of variable names and their definitions is presented. In this list, the term "plot coordinates" refers to coordinates, in inches, relative to the physical dimensions of the plotting surface, whereas "body coordinates" refers to values expressed in terms of the coordinate system in which the actual body being modeled is defined, in arbitrary units. Bear in mind that these definitions apply only to routines in which /CAL/modifications were made; similar variable names having totally different meanings may be present in the unmodified routines.

ACRIT -- Array containing the angles formed by the lower-left and upper-right corners of the envelopes surrounding node label numbers to be plotted; a function of the number of digits as well as the character height, width, and spacing, the ACRIT values are used to determine the best location at which to place the label.

BXMAX } -- Maximum and minimum nodal point X- and Y-values, in
 BXMIN } body coordinates, of the entire body being modeled; used
 BYMAX }
 BYMIN } to scale the final "BODY" plot generated by QMESH.

HCHAR -- Character height, converted to equivalent body coordinate units, of the node label numbers to be plotted; used with ACRIT to determine the best location at which to place the label.

IEVERP-- "Ever plotted?" flag value, used to indicate whether the plotter initialization call has been made (IEVERP=1) or not (IEVERP=0).

KOMMEN } -- Alphanumeric (literal) arrays containing, at four characters
 KOMPCK } per word, a user-supplied comment to be plotted.

LABEL -- Alphanumeric (literal) variable containing a node, element, or boundary condition code number to be plotted.

LIT -- Alphanumeric (literal) array containing a character string to be plotted, e. g., a numeric label for an axis tic mark.

L PLOT } -- Logical unit number to which the plotter output is to be
LPLT } written.

NCHAR -- Typically a label or title character count.

NCHKOM -- Number of characters in the literal string contained in KOMMEN.

NCHLIT -- Number of characters in the literal string contained in LIT.

NFRAME -- Counter for the number of plotter frames generated.

NNBWK -- Number of characters in the string contained in KOMPX.

NPLBUF -- The number of words (4-byte locations) allocated to the required plotter output buffer PLTBUF; used in the plotter initialization call.

PBETWN -- The horizontal spacing, in plot coordinate inches, between consecutive frames of an output plot.

PLTBUF -- The plotter output buffer array required by the Calcomp routines; see also NPLBUF.

PLTSIZ -- The length, in plot coordinate inches, of any side of the square defining the limits for plotter pen motion; using a Calcomp 765 plotter with limits $0 \leq Y \leq 11$ inches, for example, the value of PLTSIZ should be 11.0

PMARGB } -- The margin size, in plot coordinate inches, between the edges
PMARGL } of the inner square envelope defined by the plotted coordinate
PMARGR } axes and the outer square defined by PLTSIZ; the last char-
PMARGT } acter in each variable name denotes bottom, left, right, or
top; it is within these margins that titles, comments, and
axis labels are placed.

PMB } -- Same as PMARGB or PMARGL, above.
PML }

PSPACE -- On the "MAP" plot generated by QPLOT or QPLOT8 (the last frame plotted), PSPACE is the gap, in plot coordinate inches, between the horizontal or vertical lines outlining each square section and the envelope of the section identification numbers plotted at each corner of the square.

PSYMHT -- The height, in plot coordinate inches, of the (circular) symbol plotted to represent each nodal point.

PTCSZ2 -- Half the size, in plot coordinate inches, of the tic marks drawn on the coordinate axes by QPLOT and QPLOT8.

PTICSZ -- The size, in plot coordinate inches, of the tic marks drawn on the coordinate axes by QMESH.

PTXTHT } -- The values in plot coordinate inches, of character height
PTXTSP } (HT) and width (WD), and intercharacter spacing (SP),
PTXTWD } for plotted text such as titles, comments, and axis names;
all these values are used to calculate proper text placement,
but only the height (HT) parameter is required by the Calcomp
routines.

PVALHT } -- Similar to PTXTHT (etc.), above, except that these sizes
PVALSP } are used for node and element labels, "MAP" envelope labels,
PVALWD } and axis tic mark identification.

PXMAX } -- On the "MAP" plot generated by QPLOT or QPLOT8 (the
PXMIN } last frame plotted), these are the X-values, in plot coordinate
inches, of the right and left vertical lines of the "section"
envelope box being plotted.

PXSIZ -- The length, in plot coordinate inches, of the X-axis; determined
by PLTSIZ less the left and right margins.

PYMAX } -- Y-values of the top and bottom horizontal lines corresponding
PYMIN } to PXMAX and PXMIN.

PYSIZ -- The length, in plot coordinate inches of the Y-axis; determined
by PLTSIZ less the top and bottom margins.

RASD } -- Parameters used to calculate (D), and specify (DX, DY) to
 RASDX }
 RASDY } subroutine SCALR, an approximate fraction of total plot width
 at which axis tic marks are to be positioned.

SPCHAR -- Intercharacter spacing, converted to equivalent body
 coordinates, corresponding to HCHAR.

SYMHT -- Same as PSYMHT.

TITLE -- Alphanumeric (literal) array containing, at four characters
 per word, a program-generated title for each frame plotted.

WCHAR -- Character width, converted to equivalent body coordinates,
 corresponding to HCHAR.

XDEL -- Interval, in body coordinates, between adjacent tic marks
 on the X-axis.

XL -- X-value, in body coordinates, selected by the program to be
 the left side of the X-axis; this value is typically "padded" by
 a small amount over the actual nodal point coordinates to
 avoid plotting any portion of the region of interest directly
 on the plot axes.

XMAX } -- Maximum and minimum X-values, in body coordinates, of
 XMIN } all the nodal points.

XR -- X-value, in body coordinates, of the right side of the X-axis;
 its calculation is similar to XL.

XSCL -- Multiplicative scale factor used to convert X-values from body
 coordinates into plot coordinates.

YB -- Y-value, in body coordinates, of the bottom of the Y-axis;
 its calculation is similar to XL.

YDEL -- Interval, in body coordinates, between adjacent tic marks
 on the Y-axis.

YMAX } -- Maximum and minimum Y-values, in body coordinates, of all
 YMIN } the nodal points.

18

YT -- Y-value, in body coordinates, of the top of the Y-axis;
its calculation is similar to XL.

YSCL -- Multiplicative scale factor used to convert Y-values from
body coordinates into plot coordinates.

ZL -- X-value, in body coordinates, of the left-most tic mark
on the X-axis.

ZB -- Y-value, in body coordinates, of the lowest tic mark
on the Y-axis.

The conversion of coordinate data from body coordinates (X, Y) to plot coordinates (PX, PY), in inches, is performed in the following way:

$$PX = (X - XL) * XSCL + PMARGL \text{ and } PY = (Y - YB) * YSCL + PMARGB$$

where XL, YB, XSCL, YSCL, PMARGL, and PMARGB are as defined above.

The converted label placement calculations are made in terms of character height, character width, and intercharacter spacing as specified in available Calcomp documentation; for use with other plotters for which the intercharacter spacing may not be known, calculations based on width including spacing with the space parameter set to zero will probably be acceptable.

VI. Test Results and Example Problem

In the course of checking out the converted programs, a number of statistics were obtained which may be of interest, and they are therefore included in this section. The compile times and execution core requirements for both the CDC6600 and IBM360/50 versions of all five programs are listed, for comparison, in Table I. Eight test problems were used to check the performance of the converted programs against the original versions; a comparison between the execution CPU times for both versions is contained in Table II.

The last test problem listed in Table II, "Shipping Cask", is a typical cask model. The QMESH and RENUM input data decks used for the cask example are listed in Appendix B. QMESH plotted a total of 21 frames for this problem and QPLOT produced 16 frames. The complete element configuration for the cask model, as plotted by QMESH, is shown in Figure 1.

As an example of the data card setup used to define the cask, consider the upper left horizontal extension shown at the bottom of the mesh in Figure 1. The extension was defined as Regions 17 and 20, as shown in Figures 2 and 3, respectively. In these figures, the Point (P),

TABLE I

Comparison of Compilation Time and Central Memory Requirements
For the IBM and CDC Versions

Fortran Program	CPU Compile Time (Seconds)*		Core Required for Execution**	
	IBM 360/50	CDC 6600	360 Bytes (Decimal)	6600 Words (Decimal)
QMESH	374.54	50.93	250K	40.0K
RENUM	60.82	6.80	172K	37.1K
QPLOT	80.52	10.86	160K	30.2K
RENUM8	107.82	9.10	162K	33.4K
QPLOT8	85.17	8.68	186K	36.7K

*Fortran source only -- does not include Assembler or Plotter Library Routines.

**Includes Assembler, Plotter Library, and System Library Routines.

TABLE II

Comparison of Test Problem Run Times for the IBM and CDC Versions

Problem Name	QMESH/RENUM/QPLOT Execution CPU Seconds		QMESH/RENUM8/QPLOT8 Execution CPU Seconds	
	IBM 360/50	CDC 6600	IBM 360/50	CDC 6600
Four-Region Example	134.27	15.37	266.23	7.24
QPLOT Example	233.33	32.18	266.92	33.37
S.R.D. * Scheme 1	456.07	43.19	439.50	41.73
S.R.B. * Scheme 2	348.26	34.33	441.41	37.57
S.R.B. * Scheme 3	1243.76	131.91	1238.21	138.12
Notched Specimen	103.07	11.73	181.09	16.15
Kirsch Problem	62.60	8.90	111.75	11.08
Shipping Cask	1078.98	60.15	----	----

*Single Region Body -- Three different Scheme cards applied against the same geometry.

Figure 1. QMESH Plot of Shipping Cask Mesh

Figure 2. Annotated QMESH Plot of Shipping Cask Region 17

Figure 3. Annotated QMESH Plot of Shipping Cask Region 20

CARD	COLUMNS	1	2	3	4	5
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
1:	POINT	904	9.	12.5		
2:	POINT	905	0.	10.5		
3:	POINT	908	20.5	12.5		
4:	POINT	910	20.5	10.5		
5:	LINE	97	905	904	4	
6:	LINE	98	904	908	10	115
7:	LINE	913	905	910	10	
8:	LINE	914	908	910	4	
9:	REGION	1	17	-913	-914	-98 -97

NAME	COLUMNS	1	2	3	4	5
NUM960	1234567890123456789012345678901234567890123456789012345					
1:	POINT	906	22.75	10.5		
2:	POINT	907	22.5	14.5		
3:	POINT	908	20.5	12.5		
4:	POINT	909	20.5	14.5		
5:	POINT	910	20.5	19.5		
6:	POINT	911	23.25	14.5		
7:	POINT	912	23.25	10.5		
8:	LINE CIRC	90	908	907	909	114
9:	LINE	914	908	910		4
10:	LINE	915	906	912		2
11:	LINE	918	911	907		2
12:	LINE	919	906	911		3
13:	LINE	920	911	912		7
14:	LINE	93	915	919		
15:	REGION	1	20	93	-914	-908 -918 -920

EXAMPLE SHIPPING CASK

Figure 4. QPLOT Plot of Shipping Cask Section 1

Figure 5. QPLOT Plot of Shipping Cask Section 11

Figure 6. QPLOT Plot of Shipping Cask Section 12

Figure 7. QPLOT Map Showing Locations of Plotted Sections

VII. Conclusion

The final product of the conversion effort, along with this report, is a 9-track magnetic tape containing eleven EBCDIC card image files, as follows:

<u>File No.</u>	<u>Contents</u>	<u>No. of Cards</u>
1	QMESH source (Fortran)	6049
2	RENUM source (Fortran)	1109
3	QPLOT source (Fortran)	1453
4	RENUM8 source (Fortran)	1471
5	QPLOT 8 source (Fortran)	1458
6	CORE source (Assembler)	62
7	ICLOCK source (Assembler)	99
8	IDAY source (Assembler)	71
9	TIME source (Assembler)	36
10	ABEND source (Assembler)	9
11	Four-Region Example (data deck)	33

The card-image tape has been transferred to the appropriate personnel at Oak Ridge National Laboratory for implementation on their IBM computer(s). The converted codes have also been released to Argonne Code Center, from which they will be generally available.

Now that the conversion project has been completed, it seems appropriate to present some subjective observations. The Fortran modifications required to satisfy the IBM and Calcomp requirements were time-consuming but straightforward. By far the greatest difficulties arose as a result of unfamiliarity with the complexities of the IBM operating system and its Job Control Language (JCL). For conversions of the sort reported herein, it appears that for greatest efficiency the work should be performed directly on the destination mainframe by personnel who are experienced in its use and have convenient access to it. Although this recommendation is intended to apply to all computer models, it appears to be particularly appropriate in the case of IBM360 and 370 series mainframes.

References

1. R. E. Jones, "Users Manual for QMESH, a Self-Organizing Mesh Generation Program," SLA-74-0239, Sandia Laboratories, Albuquerque, New Mexico, July 1974.
2. R. E. Jones, "QMESH: A Self-Organizing Mesh Generation Program," SLA-73-1088, Sandia Laboratories, Albuquerque, New Mexico, July 1974.
3. R. E. Jones, "Improvements and Correction of Bugs in QMESH and RENUM," Memorandum to Users of QMESH/RENUM dated May 13, 1976.
4. Program and Budget Proposal Schedule 189, "Structural Technology for Shipping Containers," submitted to NRC by Sandia Laboratories, September 1975.
5. SAFER Program Brief -- Transportation, "Structural Technology for Shipping Containers," submitted to Sandia Laboratories by NRC, December 1975.
6. S. E. Benzley, "A Static Finite Element Structural Analysis System for Nuclear Regulatory Commission Cask Analysis," correspondence with W. A. Von Riesemann, Sandia Laboratories, Albuquerque, New Mexico, April 1976.
7. A. F. Schknde, "Trip Report: (1) Oak Ridge National Laboratory....; (2) U. S. Nuclear Regulatory Commission....," Memorandum to W. A. Von Riesemann, Sandia Laboratories, Albuquerque, New Mexico, dated April 29, 1976.
8. Computer Users Guide, Fourth Edition, March 1, 1976; Division of ADP Support, Office of Administration, U. S. Nuclear Regulatory Commission, Bethesda, Maryland.
9. Programmer's Notebook, December 10, 1975; Systems Support Department, Nuclear Division, Union Carbide, Oak Ridge, Tennessee.
10. IBM System/360 Operating System FORTRAN IV (G and H) Programmers Guide, Fifth Edition, September 1973. Order No. GC28-6817-4; International Business Machines Corporation, New York, N. Y.
11. IBM System/360 and System 370 FORTRAN IV Language, Eleventh Edition, May 1974. Order No. GC28-6515-10; International Business Machines Corporation, New York, N. Y.
12. FORTRAN Extended Reference Manual, 6000 Version 3, 7600 Version 1, Revision D. February 1973, Publication No. 60329100; Control Data Corporation, Sunnyvale, California.
13. SCOPE Reference Manual, 6000 Version 3.3, Revision C, July 1971, Publication No. 60305200; Control Data Corporation, Sunnyvale, California.
14. UPDATE Reference Manual, Revision F, June 1975, Publication No. 60342500; Control Data Corporation, Arden Hills, Minnesota.
15. N. A. Smith, "SCOPE Users Manual for the Control Data 6600," SC-M-70-185, Sandia Laboratories, Albuquerque, New Mexico, November 1975.
16. SC-4020 Usage with IBM-7090/7094, CDC-3600, Univac - 1107/1108, CDC-6600, SC-M-70-68, Sandia Laboratories, Albuquerque, New Mexico, March 1970.
17. Programming Calcomp Pen Plotters, September 1969, Publication No. 1006A-10M-1170; California Computer Products, Inc., Anaheim, California.
18. W. R. Garner, Jr., "Characteristics of Fortran," Martin Mariella Corporation, January 1971; available as NASA Tech Brief 73-10322, October 1973, from the Technology Utilization Office, NASA Langley Research Center, Hampton, Virginia.
19. J. W. Marchak, "FORTCOM" FORTRAN Comparison Chart, February 1976; Computer Center, The University of Texas, Austin, Texas.

APPENDIX A

IBM Deck Setups Used on the ERDA/ALO IBM360/50 During the Program Conversion Effort

- A-1 Create LIBRARIAN Files of the Fortran Programs
- A-2 Create LIBRARIAN Files of the Assembler Subroutines
- A-3 Assemble the Assembler Subroutines
- A-4 Compile and Link-Edit QMESH
- A-5 Compile and Link-Edit RENUM
- A-6 Compile and Link-Edit QPLOT
- A-7 Compile and Link-Edit RENUM8
- A-8 Compile and Link-Edit QPLOT8
- A-9 Execute QMESH/RENUM/QPLOT for Four-Region Example
- A-10 Execute QMESH/RENUM8/QPLOT8 for Four-Region Example
- A-11 Copy Card Images to Tape and Check-Print

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

1: //UR111110 JOB (SLA,NRC),'SCHKADE-CATALOG TAPE',CLASS=E
2: //STEPA EXEC LIBR
3: //LIBR.OSJOB DD DUMMY
4: //INTAPE DD LABEL=(1,NL,EXPDT=98000),DSN=S.SACNET,
5: // DISP=OLD,UNIT=TAPE7,VOL=(,RETAIN,SER=NRC01),
6: // DCB=(DEN=2,LRECL=80,RECFM=FB,BLKSIZE=5120,TRTCH=ET)
7: -OPT LIST
8: -ADD R99911,SEQ=/81,6,10,10/
9: -PGMR SLA/SCHKADE
10: -DESC QMESH/NRC--IBM VERSION
11: -AUX INTAPE
12: -EMOD
13: -END
14: //STEPB EXEC LIBR
15: //LIBR.OSJOB DD DUMMY
16: //INTAPE DD LABEL=(2,NL,EXPDT=98000),DSN=S.SACNET,
17: // DISP=OLD,UNIT=TAPE7,VOL=(,RETAIN,SER=NRC01),
18: // DCB=(DEN=2,LRECL=80,RECFM=FB,BLKSIZE=5120,TRTCH=ET)
19: -OPT LIST
20: -ADD R99912,SEQ=/81,6,10,10/
21: -PGMR SLA/SCHKADE
22: -DESC RENUM/NRC--IBM VERSION
23: -AUX INTAPE
24: -EMOD
25: -END
26: //STEPB EXEC LIBR
27: //LIBR.OSJOB DD DUMMY
28: //INTAPE DD LABEL=(3,NL,EXPDT=98000),DSN=S.SACNET,
29: // DISP=OLD,UNIT=TAPE7,VOL=(,RETAIN,SER=NRC01),
30: // DCB=(DEN=2,LRECL=80,RECFM=FB,BLKSIZE=5120,TRTCH=ET)
31: -OPT LIST
32: -ADD R99913,SEQ=/81,6,10,10/
33: -PGMR SLA/SCHKADE
34: -DESC QPLOT/NRC--IBM VERSION
35: -AUX INTAPE

```

A-1. Create LIBRARIAN Files of the Fortran Programs

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

36: -FMOD
37: -END
38: //STEPR EXEC LIRP
39: //LIRP.OSJOB DD DUMMY
40: //INTAPE DD LABEL=(4,ML,EXPDT=98000),DSN=S.SACNET.
41: // DISP=OLD,UNIT=TAPE7,VOL=(,RETAIN,SEP=NRC01),
42: // DCR=(DEN=2,LRECL=80,RECFM=FB,BLKSIZE=5120,TRTCH=FT)
43: -OPT LIST
44: -ADD P99914,SEQ=/81,6,10,10/
45: -PGMR SLA/SCHKAD
46: -DESC RENUMA/NPC--IBM VERSION
47: -AUX INTAPE
48: -FMOD
49: -END
50: //STEPR EXEC LIRP
51: //LIRP.OSJOB DD DUMMY
52: //INTAPE DD LABEL=(5,ML,EXPDT=98000),DSN=S.SACNET.
53: // DISP=OLD,UNIT=TAPE7,VOL=(,RETAIN,SEP=NRC01),
54: // DCR=(DEN=2,LRECL=80,RECFM=FB,BLKSIZE=5120,TRTCH=FT)
55: -OPT LIST
56: -ADD R99915,SEQ=/81,6,10,10/
57: -PGMR SLA/SCHKAD
58: -DESC QPLOTA/NRC--IBM VERSION
59: -AUX INTAPE
60: -FMOD
61: -END
62: /*
63: //

```

A-1. Create LIBRARIAN Files of the Fortran Programs (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
1:	//UR22222D JOB (SLA,NRC),'SCHKADE-CAT ASM.CROS',CLASS=E								
2:	//STEPA EXEC LIBR								
3:	//LIBR.DSJOB DD DUMMY								
4:	-OPT LIST								
5:	-ADD R99921,SEQ=/81,6,10,10/								
6:	-PGMR SLA/SCHKADE								
7:	-DESC CORE/ORNL--								
8:	CORE TITLE 'SUBROUTINE--FORMATTED READ/WRITE FROM/TO IN-CORE AREA'								
9:	*****								
10:	*								
11:	* IBM ASSEMBLER SUBROUTINE CORE *								
12:	* DOCUMENTED IN THE ORNL NEW PROGRAMMER'S NOTEBOOK (CH. 20, PG. 4) *								
13:	* PROVIDED BY J. A. CARPENTER (OAK RIDGE NATIONAL LABORATORY) JUNE 76 *								
14:	*								
15:	*****								
16:	CORE	CSECT	CALL CORE(BUFF,80) INTEGER*4 BUFF(20)						
17:		USING CORE,15							
18:		B CORE*12							
19:		DC X'07000000'							
20:		DC C'CORE'							
21:		STM 14,4,12(13)	SAVE REGISTERS						
22:		LM 2,3,0(1)	ADDRESS OF FIRST ARG IN R(2)						
23:		L 3,0(3)	VALUE OF SECOND ARG IN R(3)						
24:		STM 2,3,ARGS	STORE ABOVE TWO ITEMS IN ARGS						
25:		LA 1,EXIT							
26:		LA 3,L2							
27:		HALR 2,3	BRANCH TO L2						
28:	L0	LM 14,4,12(13)	RESTORE REGISTERS						
29:		SR 15,15	SET ZERO CONDITION CODE						
30:		BR 14	RETURN						
31:		USING EXIT,1							
32:	EXIT	ST 4,SAVE4							
33:		LR 4,1							
34:		DROP 1							
35:		USING EXIT,4							

A-2. Create LIBRARIAN Files of the Assembler Subroutines

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
36:		LR	1,0						
37:		TM	1(1),X'0F'						
38:		B0	F1						
39:		L	1,=V(FI0CS#)						
40:		LA	3,L2						
41:		BALR	2,3						
42:		LM	2,3,ARGS						
43:		B	E2						
44:	E1	LM	2,3,ARGS						
45:		MVI	0(2),X'40'						
46:		BCTR	3,0						
47:		BCTR	3,0						
48:		EX	3,MOVE						
49:		LA	3,2(3)						
50:	E2	L	4,SAVE4						
51:		LR	1,0						
52:		B	6(1)						
53:		DRDP	4						
54:	MOVE	MVC	1(1,2),0(2)						
55:		USING	L2,3						
56:	L2	ST	15,SAVE15						
57:		L	15,=V(IRCOM#)						
58:		MVI	74(15),X'50'						
59:		FX	0,74(15)						
60:		MVI	74(15),X'58'						
61:		L	15,SAVE15						
62:		BR	2						
63:	ARGS	DS	0F						
64:	ADDRARG1	DS	1F						
65:	ARG2	DS	1F						
66:	SAVE15	DS	1F						
67:	SAVE4	DS	1F						
68:		LTORG							
69:		END							
70:	-FM00								

SUBTRACT TWO FROM
 SECOND ARGUMENT
 MOVE (SECOND ARG - 1) CHARACTERS RIGHT 1
 INCREMENT SECOND ARGUMENT BY TWO
 BRANCH TO L0 OR L1

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

71: -END
72: //STEPB EXEC LIBR
73: //LIBR.OSJOB DD DUMMY
74: -OPT LIST
75: -ADD R99922,SEQ=/81,6,10,10/
76: -PGMR SLA/SCHKANE
77: -DESC ICLOCK/ORNL--
78: CLOC TITLE 'FUNCTION--RETURN COMPUTER TIMER READING (CENTI-SECONDS)'
79: *****
80: *
81: * IBM ASSEMBLER FUNCTION ICLOCK/ICLOCKF/ITICK
82: * DOCUMENTED IN THE ORNL NEW PROGRAMMER'S NOTEBOOK (CH. 20, PG. 7)
83: * PROVIDED BY J. A. CARPENTER (OAK RIDGE NATIONAL LABORATORY) JUNE 76
84: *
85: *****
86: *** SUBPROGRAM IKLOCK... FORTRAN FUNCTION FOR TELLING TIME
87: *** WORKS LIKE ICLOCK EXCEPT THAT IT RUNS FOR TASK-TIME (CPU TIME)
88: *** ONLY, NOT REAL-TIME
89: ***
90: *** IT=ICLOCK(0) GIVES ZERO THE FIRST TIME CALLED---
91: *** THEREAFTER, IT=IKLOCK(0) GIVES ELAPSED TIME
92: *** IN 0.01 SECOND UNITS SINCE THE FIRST CALL---
93: *** (TIME IS CUMULATIVE)
94: ***
95: *** ENTRY-POINT ITICK CAN ALSO BE USED WHICH GIVES ONLY
96: *** THE ELAPSED CPU-TIME SINCE THE IMMEDIATELY PREVIOUS
97: *** CALL
98: *** MAXIMUM TIME USABLE IS ABOUT 15.5 HOURS(TASK TIME)
99: *** IMPORTANT CONSTANTS...
100: *** 'TIMER-UNIT' IS 26 MICROSECONDS
101: *** SECONDS PER DAY= 86400
102: *** 6 HOURS IN THE TIMER=X'62F21EC0'
103: *** (1/300)/256= 13020833 (TU IS REALLY 26.141666 MIKES)
104: *** (2**32)-1 = 4294967295
105: *** (2**31)-1 = 2 147 483 648

```

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								

```

106: ***          ((2**31)-1)/384= 5592405
107: ***          384=(300*256)/(2*100)
108: ***          X'FFFFFFFF' = 4294967295
109: ***  CONVERSION ALGORITHM TAKEN FROM MODULE IEAQRTO9 (TIME SVC)
110: ***          WRITTEN BY W.B.GARDNER  APRIL 4 1969
111: ***
112: ICLOCK  START  0
113: ICLOCKF EQU *
114: ENTRY ICLOCKF
115: ENTRY ITICK
116: BC      15,12(15)
117: DC      X'07'
118: DC      C'ICLOCK'
119: STM      14,5,12(13)
120: BALR     5,8
121: USING    *,5
122: LR       4,13
123: LA       13,SAVE
124: ST       13,8(0,4)
125: ST       4,4(0,13)  CHAINS THE SAVE-AREAS
126: TM      INIT,X'FF'
127: BO      NORMAL
128: OI      INIT,X'FF'  FIRST-TIME CODING GOES IN HERE
129: TTIMER
130: LTR      0,8
131: BZ      NOTSET
132: ***          IF WE GOT HERE,SOMEONE'S ALREADY DONE A STIMER---
133: ***          WE'LL JUST USE THEIRS
134: ST       0,LONGTIME
135: SR       0,0
136: B       GETOUT
137: NOTSET DS 0H      WE MUST SET THE TIMER SOAS TO LATER DO TTIME
138: STIMER   TASK,TUINVL=LONGTIME
139: SR       0,0
140: B       GETOUT

```

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
141:	NORMAL	TTIMER							
142:	SR	2,2							
143:	L	3, LONGTIME	SHOULD ALWAYS BE POSITIVE						
144:	SR	3,0							
145:	M	2, FACTER	SCALING FACTOR						
146:	D	2, DIVIDER	CONVERT TO .01 SECOND UNITS (FROM TUI)						
147:	LR	0,3							
148:	***	R3 HAS TIME FROM START							
149:	TM	ISW, X'FF'							
150:	BZ	GETOUT							
151:	***	IF HERE WE MUST CALCULATE ITICK							
152:	S	0, LASTTIME							
153:	***	LASTTIME HAS TIME REMAINING LAST CALL							
154:	ST	3, LASTTIME							
155:	GETOUT	L 13, SAVE+4							
156:	MVI	ISW, X'00'							
157:	LM	2, 5, 28(13)							
158:	L	14, 12(13)							
159:	MVI	12(13), X'FF'	SIGNIFY THAT WE EXITED						
160:	BCR	15, 14							
161:	INIT	DC X'00'							
162:	SAVE	DS 10F							
163:	LASTTIME	DC F'0'							
164:	LONGTIME	DC X'7FFFFFFF'	WE WANT TIME REMAINING TO ALWAYS LOOK POSITIVE						
165:	FACTER	DC F'100000'							
166:	DIVIDER	DC X'0249F000'	RELEASE 17 VERSION OF THE DIVIDER						
167:	ITICK	DS 0F							
168:	BC	15, 16(15)							
169:	DC	X'07'							
170:	DC	C'ITICK'							
171:	ISW	DC X'00'							
172:	DC	AIICLOCK							
173:	OI	11(15), X'FF'	TURN ON ISW TO INDICATE A CALL TO ITICK						
174:	E	15, 12(15)	GO TO IKLOCK						
175:	BR	15							

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
176:	END								
177:	-FMODE								
178:	-END								
179:	//STEPC EXEC LIRR								
180:	//LIBR.OSJOB DD DUMMY								
181:	-OPT LIST								
182:	-ADD R99923,SEQ=/81.6,10,10/								
183:	-PGMR SLA/SCHWAE								
184:	-DESC IDAY/ORNL--								
185:	IDAY TITLE 'SUBROUTINE--RETURN CURRENT DATE AS MM-DD-YY'								
186:	*****								
187:	*								
188:	* IBM ASSEMBLER SUBROUTINE IDAY								
189:	* DOCUMENTED IN THE ORNL NEW PROGRAMMER'S NOTEBOOK (CH. 20, PG. 8)								
190:	* PROVIDED BY J. A. CARPENTER (OAK RIDGE NATIONAL LABORATORY) JUNE 76								
191:	*								
192:	*****								
193:	IDAY START 0				IDAY(DAY)	DAY IS 3 BYTES		IDAY	10
194:	USING *,15							IDAY	20
195:	STM 2,4,SAVE							IDAY	30
196:	LR 4,15							IDAY	40
197:	DROP 15							IDAY	50
198:	USING IDAY,4							IDAY	60
199:	L 2,0(0,1)				X2=A(DAY)			IDAY	70
200:	SR 1,1				X1=0 FOR CALL OF DATE			IDAY	80
201:	SVC 11				GET DATE IN X1	XX66122F		IDAY	90
202:	S 1,=F'3'				MAKE SIGN A +			IDAY	100
203:	SLL 1,8					66122C00		IDAY	110
204:	SR 0,0				ZERO X0	00000000		IDAY	120
205:	SLDL 0,8					00000066122C0000		IDAY	130
206:	SRL 1,16					000000660000122C		IDAY	140
207:	SLL 0,4					00000660		IDAY	150
208:	ST 0,DUR+4							IDAY	160
209:	OT DUB+7,X'0F'				SET SIGN TO ZONE			IDAY	170
210:	UNPK DATE+6(2),DUR+6(2)				SET YEAR IN DATE			IDAY	180

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD NUMBER	COLUMNS	1	2	3	4	5	6	7	8
211:		NI	DUB+7,X'FC'		SET SIGN TO +			IDAY	190
212:		CVB	0,DUB					IDAY	200
213:		ST	1,DUB+4					IDAY	210
214:		SR	1,1		ZERO X1			IDAY	220
215:		SRDL	0,2					IDAY	230
216:		LTR	1,1					IDAY	240
217:		BZ	LEAP					IDAY	250
218:		L	1,=F'28'					IDAY	260
219:		B	STORE					IDAY	270
220:	LEAP	L	1,=F'29'					IDAY	280
221:	STORE	ST	1,MONTH+4					IDAY	290
222:		CVB	1,DUB					IDAY	300
223:		SR	3,3		ZERO X3			IDAY	310
224:	AGAIN	S	1,MONTH(3)					IDAY	320
225:		BC	12,FOUND					IDAY	330
226:		A	3,=F'4'					IDAY	340
227:		B	AGAIN					IDAY	350
228:	FOUND	A	1,MONTH(3)		X1=DAY OF MONTH			IDAY	360
229:		SRL	3,2					IDAY	370
230:		LA	3,1(0,3)		X3=MONTH OF YEAR			IDAY	380
231:		CVD	3,DUB					IDAY	390
232:		OI	DUB+7,X'OF'		SET SIGN TO ZONE			IDAY	400
233:		UNPK	DATE(2),DUB+6(2)		SET MONTH IN DATE			IDAY	410
234:		CVD	1,DUB					IDAY	420
235:		OI	DUB+7,X'OF'		SET SIGN TO ZONE			IDAY	430
236:		UNPK	DATE+3(2),DUB+6(2)		SET DAY IN DATE			IDAY	440
237:		MVC	0(8,2),DATE		MOVE DATE INTO GIVEN ADDRESS			IDAY	450
238:		LM	2,4,SAVE					IDAY	460
239:		BR	14					IDAY	470
240:	DUB	DC	0'0'					IDAY	480
241:	MONTH	DC	F'31'	JAN				IDAY	490
242:		DC	F'00'	FEB				IDAY	500
243:		DC	F'31'	MAR				IDAY	510
244:		DC	F'30'	APR				IDAY	520
245:		DC	F'31'	MAY				IDAY	530

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
246:		DC	F'30'	JUN				IDAY	540
247:		DC	F'31'	JUL				IDAY	550
248:		DC	F'31'	AUG				IDAY	560
249:		DC	F'30'	SEP				IDAY	570
250:		DC	F'31'	OCT				IDAY	580
251:		DC	F'30'	NOV				IDAY	590
252:		DC	F'31'	DEC				IDAY	600
253:	DATE	DC	C 00-10-00					IDAY	610
254:	SAVE	DS	3F					IDAY	620
255:		END						IDAY	630
256:	-EMOD								
257:	-END								
258:	//STEP0	FXEC	LIBR						
259:	//LIBR.OSJOB	DD	DUMMY						
260:	-OPT LIST								
261:	-ADD R99924,SEQ=781,6,10,10/								
262:	-PGMR SLA/SCHKADE								
263:	-DESC TIME/ORNL--								
264:	TIME	TITLE	'SUBROUTINE--RETURN CURRENT WALL-CLOCK TIME AS HH.MM.SS'						
265:	*****								
266:	*								
267:	* IBM ASSEMBLER SUBROUTINE TIME								
268:	* DOCUMENTED IN THE ORNL NEW PROGRAMMER'S NOTEBOOK (CH. 20, PG. 12)								
269:	* PROVIDED BY J. A. CARPENTER (OAK RIDGE NATIONAL LABORATORY) JUNE 76								
270:	*								
271:	*****								
272:	TIME	START	0						
273:		USING	*,12						
274:		STM	14,12,12(13)						
275:		LR	12,15						
276:		ST	13,SAVE						
277:		LA	2,SAVE						
278:		ST	2,8(0,13)						
279:		L	5,0(1) ADDR OF ARG						
280:		TIME							

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	12345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901
281:		ST	0,TIMEHOLD						
282:		OI	TIMEHOLD+3,B'00001111'						
283:		UNPK	TIMEHOLD(7),TIMEHOLD(4)						
284:		LA	3,TIMEODAY+7						
285:		LA	2,3						
286:		LA	4,TIMEHOLD						
287:		LA	1,TIMEODAY						
288:	LOOP	MVC	0(2,1),0(4)						
289:		LA	4,2(4)						
290:		BXLE	1,2,LOOP						
291:		MVC	0(8,5),TIMEODAY						
292:		L	13,SAVE						
293:		LM	14,12,12(13)						
294:		MVI	12(13),X'FF'						
295:		BR	14						
296:	TIMEODAY	DC	C' . . . '						
297:	TIMEHOLD	DS	10						
298:	SAVE	DS	1F						
299:		END							
300:	-ENOD								
301:	-END								
302:	//STEPE	EXEC	LIBR						
303:	//LIBR.OSJOB	DD	DUMMY						
304:	-OPT	LIST							
305:	-ADD	R99925,SEQ=	/81,6,10,10/						
306:	-PGMR	SLA/SCHKADE							
307:	-DESC	ABEND/PETRIE(ORNL)--							
308:	ABND	TITLE	'SUBROUTINE--INDUCE AN ABEND (ABNORMAL TERMINATION)'						
309:	*****								
310:	* IBM ASSEMBLER	SUBROUTINE	ABEND						*
311:	* PROVIDED BY	LESTER PETRIE	(OAK RIDGE NATIONAL LABORATORY) AUGUST 76 *						
312:	*****								
313:	ABEND	START							
314:		USING	ABEND,15						
315:		ABEND	3333,DUMP,STEP						

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER		1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

316:      END
317: -EMOD
318: -END
319: /*
320: //

```

A-2. Create LIBRARIAN Files of the Assembler Subroutines (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
11	//UR33333D JOB (SLA,NRC),'SCHKADE-ASM ORNL CDS',CLASS=E								
21	//S1 EXEC ASSEMBLY,LPARM='XREF,LIST,NCAL',NAME=CORE								
31	-OPT LIST								
41	-SEL R99921,PPPZ								
51	-EMOD								
61	-END								
71	/*								
81	//LKED.SYSLMOD DD DSN=NRCFILE(CORE),DISP=SHR								
91	//S2 EXEC ASSEMBLY,LPARM='XREF,LIST,NCAL',NAME=ICLOCK								
101	-OPT LIST								
111	-SEL R99922,JFMW								
121	-EMOD								
131	-END								
141	/*								
151	//LKED.SYSLMOD DD DSN=NRCFILE(ICLOCK),DISP=SHR								
161	//S3 EXEC ASSEMBLY,LPARM='XREF,LIST,NCAL',NAME=IDAY								
171	-OPT LIST								
181	-SEL R99923,HJLZ								
191	-EMOD								
201	-END								
211	/*								
221	//LKED.SYSLMOD DD DSN=NRCFILE(IDAY),DISP=SHR								
231	//S4 EXEC ASSEMBLY,LPARM='XREF,LIST,NCAL',NAME=TIME								
241	-OPT LIST								
251	-SEL R99924,ZCBX								
261	-EMOD								
271	-END								
281	/*								
291	//LKED.SYSLMOD DD DSN=NRCFILE(TIME),DISP=SHR								
301	//S5 EXEC ASSEMBLY,LPARM='XREF,LIST,NCAL',NAME=ABEND								
311	-OPT LIST								
321	-SEL R99925,KXKF								
331	-EMOD								
341	-END								
351	/*								
361	//LKED.SYSLMOD DD DSN=NRCFILE(ABEND),DISP=SHR								
371	//								

A-3. Assemble the Assembler Subroutines

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

1: //UR444440 JOB (SLA,NRC),'SCHKADE-COMP QMESH',REGION=279K,CLASS=A
2: //STEP1 EXEC LIAR
3: //SYSIN DD *
4: -OPT LIST
5: -SEL R99911,XGMP
6: -EDIT /*/(/
7: -EDIT /(/(/
8: -EDIT /*/=/
9: -EDIT /*/+//
10: -EDIT /*/;/
11: -EMOD X
12: -END
13: //STEP2 EXEC FORTHCL,PARM.FORT='XREF,MAP,ID,LINECNT=60',TIME=230,
14: // REGION=270K
15: //FORT.SYSUT2 DD UNIT=DISK,SPACE=(CYL,10)
16: //FORT.SYSIN DD DSN=*,STEP1.LIAR,DSJOB,DISP=(OLD,DELETE)
17: //LKED.SYSLIB DD DSN=NPCFILE,DISP=SHR
18: //          DD DSN=SYS1.FORTLIB,DISP=SHR
19: //          DD DSN=ERDA.LINKLIB,DISP=SHR
20: //LKED.SYSLMOD DD DSN=NRCFILE,DISP=OLD
21: //LKED.SYSIN DD *
22: INCLUDE SYSLIB(CORE)
23: INCLUDE SYSLIB(ICLOCK)
24: INCLUDE SYSLIB(IDAY)
25: INCLUDE SYSLIB(TIME)
26: INCLUDE SYSLIB(ABEND)
27: NAME EXQMESH(R)
28: /*
29: //

```

A-4. Compile and Link-Edit QMESH

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								

```

11 //UR5555D JOB (SLA,NRC),'SCHKADE-COMP RENUM',REGION=270K,CLASS=A
21 //STEP1 EXEC LIBR
31 //SYSIN DD *
41 -OPT LIST
51 -SEL R99912,SVL8
61 -EDIT /*/(/
71 -EDIT /</(/
81 -EDIT /*/=/
91 -EDIT /&/+//
101 -EDIT /!//;
111 -EMOD X
121 -END
131 //STEP2 EXEC FORTHCL,PARM.FORT='XREF,MAP,ID,LINECNT=60',TIME=230,
141 // REGION=270K
151 //FORT.SYSUT2 DD UNIT=DISK,SPACE=(CYL,10)
161 //FORT.SYSIN DD DSN=*.STEP1.LIBR,DISP=(OLD,DELETE)
171 //LKED.SYSLIB DD DSN=NRCFILE,DISP=SHR
181 //          DD DSN=SYS1.FORTLIB,DISP=SHR
191 //          DD DSN=ERDA.LINKLIB,DISP=SHR
201 //LKED.SYSLMOD DD DSN=NRCFILE,DISP=OLD
211 //LKED.SYSIN DD *
221 INCLUDE SYSLIB(CORE)
231 INCLUDE SYSLIB(ICLOCK)
241 INCLUDE SYSLIB(IDAY)
251 INCLUDE SYSLIB(TIME)
261 INCLUDE SYSLIB(ABEND)
271 NAME EXRENUM(R)
281 /*
291 //

```

A-5. Compile and Link-Edit RENUM

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

1: //UP66666D JOB (SLA,NPG),'SCHKADF-COMD QPLOT',REGION=270K,CLASS=A
2: //STEP1 EXEC LIBR
3: //SYSIN DD *
4: -OPT LIST
5: -SFL R99913,MKWR
6: -EDIT /*//
7: -EDIT /<//
8: -EDIT /#//
9: -EDIT /&+/
10: -EDIT /!//
11: -EMOD X
12: -END
13: //STEP2 EXEC FORTHCL,PARM.FORT='XREF,MAP,IO,LINECNT=60',TIME=230,
14: // REGION=270K
15: //FORT.SYSUT2 DD UNIT=DISK,SPACE=(CYL,10)
16: //FORT.SYSIN DD DSN=*.STEP1.LIBR,DSJOB,DISP=(OLD,DELETE)
17: //LKED.SYSLIB DD DSN=NPGFILE,DISP=SHR
18: //          DD DSN=SYS1.FORTLIB,DISP=SHR
19: //          DD DSN=FRDA.LINKLIB,DISP=SHR
20: //LKED.SYSLMOD DD DSN=NPGFILE,DISP=OLD
21: //LKED.SYSIN DD *
22: INCLUDE SYSLIB(CORE)
23: INCLUDE SYSLIB(ICLOCK)
24: INCLUDE SYSLIB(IDAY)
25: INCLUDE SYSLIB(TIME)
26: INCLUDE SYSLIB(ARFNO)
27: NAME EXQLOT(R)
28: /*
29: //

```

A-6. Compile and Link-Edit QPLOT

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								

```

11 //UR77777D JOB (SLA,NRC),'SCHKADE-COMP RENUM8',REGION=270K,CLASS=A
21 //STEP1 EXEC LIOR
31 //SYSIN DD *
41 -OPT LIST
51 -SEL R99914,NRLF
61 -EDIT /*//
71 -EDIT /<>//
81 -EDIT /*/=//
91 -EDIT /&/+//
101 -EDIT /!//;
111 -EOD X
121 -END
131 //STEP2 EXEC FORTHCL,PARM.FORT='XREF,MAP,ID,LINECNT=60',TIME=230,
141 // REGION=270K
151 //FORT.SYSUT2 DD UNIT=DISK,SPACE=(CYL,10)
161 //FORT.SYSIN DD DSN=*.STEP1.LIOR.05JOB,DISP=(OLD,DELETE)
171 //LKED.SYSLIB DD DSN=NRCFILE,DISP=SHR
181 //          DD DSN=SYS1.FORTLIB,DISP=SHR
191 //          DD DSN=EROA.LINKLIB,DISP=SHR
201 //LKED.SYSLMOD DD DSN=NRCFILE,DISP=OLD
211 //LKED.SYSIN DD *
221 INCLUDE SYSLIB(CORE)
231 INCLUDE SYSLIB(ICLOCK)
241 INCLUDE SYSLIB(IDAY)
251 INCLUDE SYSLIB(TIME)
261 INCLUDE SYSLIB(AREND)
271 NAME EXRENUM8(R)
281 /*
291 //

```

A-7. Compile and Link-Edit RENUM8

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
1:	//UR888880 JOB (SLA,NRC),'SCHKADE-COMP QPLOT8',REGION=270K,CLASS=A								
2:	//STEP1 EXEC LIBR								
3:	//SYSIN DD *								
4:	-OPT LIST								
5:	-SEL R99915,LSMW								
6:	-EDIT /%/(/								
7:	-EDIT /</)/								
8:	-EDIT /#/=/								
9:	-EDIT /&/+//								
10:	-EDIT /!//;/								
11:	-EMOD X								
12:	-END								
13:	//STEP2 EXEC FORTHCL.PARM.FORT='XREF,MAP,IO,LINECNT=60',TIME=230,								
14:	// REGION=270K								
15:	//FORT.SYSUT2 DD UNIT=DISK,SPACE=(CYL,10)								
16:	//FORT.SYSIN DD DSN=*.STEP1.LIBR.0SJOB,DISP=(OLD,DELETE)								
17:	//LKED.SYSLIB DD DSN=NRCFILE,DISP=SHR								
18:	// DD DSN=SYS1.FORTLIB,DISP=SHR								
19:	// DD DSN=ERDA.LINKLIB,DISP=SHR								
20:	//LKED.SYSLMOD DD DSN=NRCFILE,DISP=OLD								
21:	//LKED.SYSIN DD *								
22:	INCLUDE SYSLIB(CORE)								
23:	INCLUDE SYSLIB(ICLOCK)								
24:	INCLUDE SYSLIB(IDAY)								
25:	INCLUDE SYSLIB(TIME)								
26:	INCLUDE SYSLIB(ABEND)								
27:	NAME EXQPLOT8(P)								
28:	/*								
29:	//								

A-8. Compile and Link-Edit QPLOT8

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
1:	//UR00001D	JOB	(SLA,NRC),	'SCHKADE-EXECUTE QQQ',	REGION=300K,	CLASS=B			
2:	//STEP1	EXEC	PGM=FXQMESH						
3:	//STEPLIB	DD	DSN=NRCFILE,	DISP=SMR					
4:	//FT05F001	DD	DDNAME=SYSIN						
5:	//FT06F001	DD	SYSOUT=A						
6:	//FT07F001	DD	SYSOUT=B						
7:	//FT08F001	DD	DCB=(RECFM=VSB, BLKSIZE=6444),	UNIT=DISK,					
8:	//		SPACE=(1600,1000,RLSE),	DISP=(NEW,PASS),	DSN=	&&TEMP			
9:	//FT55F001	DD	DCB=(RECFM=VSB, BLKSIZE=6444),	UNIT=DISK,					
10:	//		SPACE=(1600,1000,RLSE),	DISP=(NEW,DELETE)					
11:	//FT66F001	DD	SYSOUT=A						
12:	//PLOT TAPE	DD	DSN=PLOT, UNIT=TAPE7, LABEL=(1,NL,,RETPD=7),	DISP=(,KEEP)					
13:	//SYSUDUMP	DD	SYSOUT=A						
14:	//SYSIN	DD	*						
15:	COMEN	EXAMPLE	BODY WITH FOUR REGIONS						
16:	POINT	1=	-1.	0.					
17:	POINT	2=	0.	0.					
18:	POINT	3=	1.	0.					
19:	POINT	4=	.470	0.					
20:	POINT	5=	.517	0.					
21:	POINT	6=	.517	.241					
22:	POINT	7=	.476	.241					
23:	POINT	8=	.448	.103					
24:	POINT	9=	-.310	.414					
25:	POINT	10=	-.345	.259					
26:	POINT	11=	-.9655	.259					
27:	LINE	1/	1	4	17				
28:	LINE	2/	4	5	2				
29:	LINE	3/	5	3	6				
30:	LINE	CIRC	4/	3	11	2	20		
31:	LINE	CIRC	5/	11	1	2	3		
32:	LINE	10/	4	8	2				
33:	LINE	11/	5	6	4				
34:	LINE	12/	6	7	2				
35:	LINE	13/	7	8	2				

A-9. Execute QMESH/RENUM/QPLOT for Four-Region Example

CARD NUMBER	COLUMNS	1	2	3	4	5	6	7	8
		12345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901
36:	LINE	14/	7	9	12				
37:	LINE	15/	9	10	2				
38:	LINE	16/	10	8	12				
39:	LINE	17/	10	11	6				
40:	SIDE	1/	10	13					
41:	SIDE	2/	11	12	14	15			
42:	REGION	1/	-2	-11	-12	1			
43:	REGION	2/	-13	-14	-15	-16			
44:	REGION	3/	-1	-10	-16	-17	-5		
45:	REGION	4/	-3	-4	-17	2			
46:	SCHEMF		JJJ	R4SRIP					
47:	END								
48:	//STEP2 EXEC	PGM=EXRENUM							
49:	//STEPLIB	DD	DSN=NRCFILE,DISP=SHR						
50:	//FT05F001	DD	DDNAME=SYSIN						
51:	//FT06F001	DD	SYSOUT=A						
52:	//FT07F001	DD	SYSOUT=B						
53:	//FT08F001	DD	DSN=&&TEMP,DISP=(OLD,DELETE)						
54:	//FT09F001	DD	DCB=(RECFM=VSB,RLKSIZE=6444),UNIT=DISK,						
55:	//	SPACE=(1680,1000,RLSE),DISP=(NEW,PASS),DSN=&&TEMP2							
56:	//FT66F001	DD	SYSOUT=A						
57:	//SYSUDUMP	DD	SYSOUT=A						
58:	//SYSIN	DD	*						
59:	END								
60:	//STEP3 EXEC	PGM=FXOPLCT							
61:	//STEPLIB	DD	DSN=NRCFILE,DISP=SHR						
62:	//FT05F001	DD	DDNAME=SYSIN						
63:	//FT06F001	DD	SYSOUT=A						
64:	//FT07F001	DD	SYSOUT=B						
65:	//FT09F001	DD	DSN=&&TEMP2,DISP=(OLD,DELETE)						
66:	//FT66F001	DD	SYSOUT=A						
67:	//PLOT TAPE	DD	DSN=PLOT,UNIT=TAPF7,LABEL=(1,NL,,,RETP7=7),DISP=(,KEEP)						
68:	//SYSUDUMP	DD	SYSOUT=A						
69:	//SYSIN	DD	DUMMY						
70:	/*								
71:	//								

A-9. Execute QMESH/RENUM/QPLOT for Four-Region Example (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
1:	//UR00002D	JOB	(SLA,NRC),	'SCHKADE-EXECUTE QRQ8',	REGION=300K,	CLASS=B			
2:	//STEP1	EXEC	PGM=EXQMESH						
3:	//STEPLIB	DD	DSN=NRCFILE,	DISP=SHR					
4:	//FT05F001	DD	DDNAME=SYSIN						
5:	//FT06F001	DD	SYSOUT=A						
6:	//FT07F001	DD	SYSOUT=B						
7:	//FT08F001	DD	DCB=(RECFM=VSB,BLKSIZE=6444),	UNIT=DISK,					
8:	//		SPACE=(1680,1000,RLSE),	DISP=(NEW,PASS),	DSN=2&TEMP				
9:	//FT55F001	DD	DCB=(RECFM=VSB,BLKSIZE=6444),	UNIT=DISK,					
10:	//		SPACE=(1680,1000,RLSE),	DISP=(NEW,DELETE)					
11:	//FT66F001	DD	SYSOUT=A						
12:	//PLOTAPPE	DD	DSN=PLOT,UNIT=TAPE7,	LABEL=(1,NL,,RETPD=7),	DISP=(,KEEP)				
13:	//SYSUDUMP	DD	SYSOUT=A						
14:	//SYSIN	DD	*						
15:	COMMEN	EXAMPLE	BODY WITH FOUR	REGIONS					
16:	POINT	1=	-1.	0.					
17:	POINT	2=	0.	0.					
18:	POINT	3=	1.	0.					
19:	POINT	4=	.470	0.					
20:	POINT	5=	.517	0.					
21:	POINT	6=	.517	.241					
22:	POINT	7=	.476	.241					
23:	POINT	8=	.448	.103					
24:	POINT	9=	-.310	.414					
25:	POINT	10=	-.345	.259					
26:	POINT	11=	-.9655	.259					
27:	LINE	1/	1	4	17				
28:	LINE	2/	4	5	2				
29:	LINE	3/	5	3	6				
30:	LINE	CIRC	4/	3	11	2	20		
31:	LINE	CIRC	5/	11	1	2	3		
32:	LINE	10/	4	8	2				
33:	LINE	11/	5	6	4				
34:	LINE	12/	6	7	2				
35:	LINE	13/	7	8	2				

A-10. Execute QMESH/RENUM8/QPLOT8 for Four-Region Example

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
36:	LINE	14/	7	9		12			
37:	LINE	15/	9	10		2			
38:	LINE	16/	10	8		12			
39:	LINE	17/	10	11		6			
40:	SIDE	1/	10	13					
41:	SIDE	2/	11	12	14	15			
42:	REGION	1/	-2	-11	-12	1			
43:	REGION	2/	-13	-14	-15	-16			
44:	REGION	3/	-1	-10	-16	-17	-5		
45:	REGION	4/	-3	-4	-17	2			
46:	SCHEME			JJJ	R(SRIP				
47:	END								
48:	//STEP2 EXEC	PGM=EXRENUM8							
49:	//STEPLIB	DD	DSN=NRCFILE,DISP=SHR						
50:	//FT05F001	DD	DDNAME=SYSIN						
51:	//FT06F001	DD	SYSOUT=A						
52:	//FT07F001	DD	SYSOUT=B						
53:	//FT08F001	DD	DSN=&&TEMP,DISP=(OLD,DELETE)						
54:	//FT09F001	DD	DCB=(RECFM=VS3,BLKSIZE=5444),UNIT=DISK,						
55:	//		SPACE=(1600,1000,RLSE),DISP=(NEW,PASS),DSN=&&TEMP2						
56:	//FT66F001	DD	SYSOUT=A						
57:	//SYSUDUMP	DD	SYSOUT=A						
58:	//SYSIN	DD	*						
59:	END								
60:	//STEP3 EXEC	PGM=EXQPL0T8							
61:	//STEPLIB	DD	DSN=NRCFILE,DISP=SHR						
62:	//FT05F001	DD	DDNAME=SYSIN						
63:	//FT06F001	DD	SYSOUT=A						
64:	//FT07F001	DD	SYSOUT=B						
65:	//FT09F001	DD	DSN=&&TEMP2,DISP=(OLD,DELETE)						
66:	//FT66F001	DD	SYSOUT=A						
67:	//PLOT TAPE	DD	DSN=PLOT,UNIT=TAPE7,LABEL=(1,NL,,RETPD=7),DISP=(,KEEP)						
68:	//SYSUDUMP	DD	SYSOUT=A						
69:	//SYSIN	DD	DUMMY						
70:	//								
71:	//								

A-10. Execute QMESH/RENUM8/QPLOT8 for Four-Region Example (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								

```

1: //UR00031D JOB (SLA,NRC),'SCHKADE-COPY TO TAPE',REGION=100K,CLASS=E
2: //COPYLIBR PROC FILE=1
3: //LIBR EXEC PGM=LIBRAFO,PARM='NJTA,NRJS'
4: //MASTER DD DSN=ERDA.LIBRMSTR,DISP=SHR
5: //BACKUP DD DUMMY
6: //SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,9LKSIZE=133)
7: //LIST DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,9LKSIZE=133)
8: //INDEX DD SYSOUT=A,DCB=(RECFM=FBA,LRECL=133,9LKSIZE=133)
9: //SYSPUNCH DD SYSOUT=B,DCB=(RECFM=FBA,LRECL=133,9LKSIZE=133)
10: //OSJOB DD DSN=CARDINGS,UNIT=TAPE9,VOL=(,RETAIN,SER=NRC03),
11: // DISP=(NEW,KEEP),DCB=(RECFM=F9,LRECL=80,8LKSIZE=4000),
12: // LABEL=(&FILE,BLP,EXPDT=98000)
13: // PEND
14: //PRINTAPE PROC FILE=1
15: //PTPCH EXEC PGM=IEBPTPCH
16: //SYSPRINT DD SYSOUT=A
17: //SYSUT1 DD DSN=CARDINGS,UNIT=TAPE9,VOL=(,RETAIN,SER=NRC03),
18: // DISP=(OLD,KEEP),DCB=(RECFM=F8,LRECL=30,8LKSIZE=4000),
19: // LABEL=(&FILE,NL,EXPDT=98000)
20: //SYSUT2 DD SYSOUT=A
21: // PEND
22: //STEP1 EXEC COPYLIBR,FILE=1
23: -OPT
24: -SEL R99911,XGMP
25: -EMOD
26: -END
27: //STEP2 EXEC COPYLIBR,FILE=2
28: -OPT
29: -SEL R99912,SVLB
30: -EMOD
31: -END
32: //STEP3 EXEC COPYLIBR,FILE=3
33: -OPT
34: -SEL R99913,NKWB
35: -EMOD

```

A-11. Copy Card Images to Tape and Check Print

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

36: -END
37: //STEP4 EXEC COPYLIBR,FILE=4
38: -OPT
39: -SEL R99914,NRLF
40: -FMOD
41: -END
42: //STEP5 EXEC COPYLIBR,FILE=5
43: -OPT
44: -SEL R99915,LSMW
45: -EMOD
46: -END
47: //STEP6 EXEC COPYLIBR,FILE=6
48: -OPT
49: -SFL R99921,PPPZ
50: -EMOD
51: -END
52: //STEP7 EXEC COPYLIBR,FILE=7
53: -OPT
54: -SFL R99922,JFMW
55: -EMOD
56: -END
57: //STEP8 EXEC COPYLIBR,FILE=8
58: -OPT
59: -SEL R99923,HJLZ
60: -FMOD
61: -END
62: //STEP9 EXEC COPYLIBR,FILE=9
63: -OPT
64: -SEL R99924,ZCRX
65: -EMOD
66: -END
67: //STEP10 EXEC COPYLIBR,FILE=10
68: -OPT
69: -SFL R99925,KXKF
70: -EMOD

```

A-11. Copy Card Images to Tape and Check-Print (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER		1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
71:	-END								
72:	//STEP11 EXFC PGM=IFBGENER								
73:	//SYSPRINT DD SYSOUT=A								
74:	//SYSUT2 DD DSN=CARDINGS,UNIT=TAPE9,VOL=(,RETAIN,SER=NRC03),								
75:	// DISP=(NEW,KEEP),DCB=(RECFM=FB,LRECL=80,BLKSIZE=4000),								
76:	// LABEL=(11,NL,EXPDT=98000)								
77:	//SYSIN DD DUMMY								
78:	//SYSUT1 DD *								
79:	COMEN								
80:	POINT	1=	-1.	0.					
81:	POINT	2=	0.	0.					
82:	POINT	3=	1.	0.					
83:	POINT	4=	.470	0.					
84:	POINT	5=	.517	0.					
85:	POINT	6=	.517	.241					
86:	POINT	7=	.476	.241					
87:	POINT	8=	.448	.103					
88:	POINT	9=	-.310	.414					
89:	POINT	10=	-.345	.259					
90:	POINT	11=	-.9655	.259					
91:	LINE	1/	1	4	17				
92:	LINE	2/	4	5	2				
93:	LINE	3/	5	3	6				
94:	LINE CIRC	4/	3	11	2	20			
95:	LINE CIRC	5/	11	1	2	3			
96:	LINE	10/	4	8	2				
97:	LINE	11/	5	6	4				
98:	LINE	12/	6	7	2				
99:	LINE	13/	7	8	2				
100:	LINE	14/	7	9	12				
101:	LINE	15/	9	10	2				
102:	LINE	16/	10	8	12				
103:	LINE	17/	10	11	6				
104:	SIDE	1/	10	13					
105:	SIDE	2/	11	12	14	15			

A-11. Copy Card Images to Tape and Check-Print (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	12345678901234567890123456789012345678901234567890123456789012345678901234567890								
106:	REGION	1/	-2	-11	-12	1			
107:	REGION	2/	-13	-14	-15	-16			
108:	REGION	3/	-1	-10	-16	-17	-5		
109:	REGION	4/	-3	-4	-17	2			
110:	SCHEME		JJJ	R(SR)P					
111:	END								
112:	//STEP21 EXEC PRINTAPE,FILE=1								
113:	//SYSIN DD *								
114:	PRINT MAXFLDS=1								
115:	TITLE ITEM=('QMESH SOURCE CARD IMAGES',90)								
116:	RECORD FIELD=(80)								
117:	//STEP22 EXEC PRINTAPE,FILE=2								
118:	//SYSIN DD *								
119:	PRINT MAXFLDS=1								
120:	TITLE ITEM=('RENUM SOURCE CARD IMAGES',90)								
121:	RECORD FIELD=(80)								
122:	//STEP23 EXEC PRINTAPE,FILE=3								
123:	//SYSIN DD *								
124:	PRINT MAXFLDS=1								
125:	TITLE ITEM=('QPLOT SOURCE CARD IMAGES',90)								
126:	RECORD FIELD=(80)								
127:	//STEP24 EXEC PRINTAPE,FILE=4								
128:	//SYSIN DD *								
129:	PRINT MAXFLDS=1								
130:	TITLE ITEM=('RENUMB SOURCE CARD IMAGES',90)								
131:	RECORD FIELD=(80)								
132:	//STEP25 EXEC PRINTAPE,FILE=5								
133:	//SYSIN DD *								
134:	PRINT MAXFLDS=1								
135:	TITLE ITEM=('QPLOTB SOURCE CARD IMAGES',90)								
136:	RECORD FIELD=(80)								
137:	//STEP26 EXEC PRINTAPE,FILE=6								
138:	//SYSIN DD *								
139:	PRINT MAXFLDS=1								
140:	TITLE ITEM=('CORE SOURCE CARD IMAGES',90)								

A-11. Copy Card Images to Tape and Check-Print (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890

```

141:   RECORD FIELD=(80)
142: //STEP27 EXEC PRINTAPE,FILE=7
143: //SYSIN   DD *
144:   PRINT MAXFLOS=1
145:   TITLE ITEM=('ICLOCK SOURCE CARD IMAGES',90)
146:   RECORD FIELD=(80)
147: //STEP28 EXEC PRINTAPE,FILE=8
148: //SYSIN   DD *
149:   PRINT MAXFLOS=1
150:   TITLE ITEM=('IDAY SOURCE CARD IMAGES',90)
151:   RECORD FIELD=(80)
152: //STEP29 EXEC PRINTAPE,FILE=9
153: //SYSIN   DD *
154:   PRINT MAXFLOS=1
155:   TITLE ITEM=('TIME SOURCE CARD IMAGES',90)
156:   RECORD FIELD=(80)
157: //STEP30 EXEC PRINTAPE,FILE=10
158: //SYSIN   DD *
159:   PRINT MAXFLOS=1
160:   TITLE ITEM=('ABEND SOURCE CARD IMAGES',90)
161:   RECORD FIELD=(80)
162: //STEP31 EXEC PRINTAPE,FILE=11
163: //SYSIN   DD *
164:   PRINT MAXFLOS=1
165:   TITLE ITEM=('EXAMPLE DATA DECK CARD IMAGES',90)
166:   RECORD FIELD=(80)
167: /*
168: //
```

A-11. Copy Card Images to Tape and Check-Print (cont)

APPENDIX B

QMESH and RENUM Input Data Decks for the Example Shipping Cask Problem

B-1. QMESH Data Deck

B-2. RENUM Data Deck

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
1:	COMMEN	EXAMPLE SHIPPING CASK							
2:	REGION	1	1	11	-13	-14	-15		
3:	POINT	101	0.		191.25				
4:	POINT	102	22.5		191.25				
5:	POINT	103	29.5		191.25				
6:	POINT	104	29.5		201.				
7:	POINT	105	0.		201.			116	
8:	LINE	11	101	102		16		108	
9:	LINE	12	102	103		8			
10:	LINE	13	103	104		8			
11:	LINE	14	104	105		24		301	
12:	LINE	15	105	101		8			
13:	SIDE	11	11	12					
14:	REGION	2	2	-21	-22	-23	-24		
15:	POINT	201	24.25		6.				
16:	POINT	202	27.0		6.				
17:	POINT	203	27.0		24.25				
18:	POINT	204	24.2		24.25				
19:	LINE	21	201	202		3			
20:	LINE	22	202	203		18			
21:	LINE	23	203	204		3			
22:	LINE	24	204	201		18			
23:	REGION	1	3	31	32	-36	-31		
24:	POINT	301	0.		4.				
25:	POINT	302	23.25		4.				
26:	POINT	303	23.25		6.				
27:	POINT	304	23.25		8.				
28:	POINT	305	22.75		8.				
29:	POINT	306	0.		8.				
30:	SIDE	31	32	33					
31:	SIDE	32	34	35					
32:	LINE	31	301	302		16		384	
33:	LINE	32	302	303		2			
34:	LINE	33	303	304		2			
35:	LINE	34	304	305		2			

B-1. QMESH Data Deck

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	12345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901	2345678901
36:	LINE		35	305	306		14		
37:	LINE		36	306	301		4		
38:	REGION	3	4	1	-44	-45	-46		
39:	SIDE		1	41	23	43			
40:	POINT		401	23.25		24.25			
41:	POINT		402	29.25		24.25			
42:	POINT		403	29.25		173.			
43:	POINT		404	23.25		173.			
44:	LINE		41	401	204		1		
45:	LINE		43	203	402		2		
46:	LINE		44	402	403		80		
47:	LINE		45	403	404		6		
48:	LINE		46	404	401		80		
49:	REGION	1	5	-52	-59	51	52		
50:	REGION	1	13	-58	-53	-54	53		
51:	POINT		501	23.25		0.			
52:	POINT		502	35.5		0.			
53:	POINT		503	35.5		15.25			
54:	POINT		504	31.0		15.25			
55:	POINT		505	31.0		6.			
56:	POINT		506	31.0		0.			
57:	LINE		51	302	501		4		
58:	LINE		52	501	506		8		305
59:	LINE		53	502	503		15		
60:	LINE		54	503	504		4		
61:	LINE		55	504	505		9		
62:	LINE		56	505	202		4		
63:	LINE		57	201	303		1		
64:	LINE		58	506	502		4		305
65:	LINE		59	506	505		6		
66:	SIDE		51	56	21	57			
67:	SIDE		52	32	51				
68:	SIDE		53	55	59				
69:	REGION	1	6	-63	61	-677	62		
70:	POINT		614	29.25		18.25			

B-1. QMESH Data Deck (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
71:	POINT	615	29.25	174.					
72:	LINE	676	605	614	3				
73:	LINE	677	614	601	4				
74:	LINE	678	606	615	3				
75:	LINE	679	615	403	1				
76:	LINE	680	402	614	5				
77:	LINE	681	512	615	1				
78:	REGION	1	14	-676	67	-678	63		
79:	REGION	1	15	-68	64	65	66		
80:	REGION	1	16	-672	-673	-45	-679	-681	
81:	POINT	601	29.25	15.25					
82:	POINT	602	41.0	15.25					
83:	POINT	603	41.0	18.25					
84:	POINT	604	40.0	18.25					
85:	POINT	605	31.25	18.25					
86:	POINT	606	31.25	174.					
87:	POINT	607	40.0	174.					
88:	POINT	608	41.0	174.					
89:	POINT	609	41.0	177.0					
90:	POINT	610	35.5	177.0					
91:	POINT	611	29.25	177.0					
92:	POINT	612	29.25	175.					
93:	POINT	613	23.25	175.					
94:	POINT	616	31.25	24.25					
95:	POINT	617	31.25	173.					
96:	LINE	61	601	504	3				
97:	LINE	62	503	602	5				
98:	LINE	63	602	603	4				
99:	LINE	64	604	605	7			103	
100:	LINE	65	605	606	86			102	
101:	LINE	66	606	607	7			101	
102:	LINE	67	607	608	2				
103:	LINE	68	608	609	4				
104:	LINE	69	609	610	5				
105:	LINE	670	610	611	7				

B-1. QMESH Data Deck (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	1234567890123456789012345678901234567890123456789012345678901234567890								
106:	LINE		671	611	612				3
107:	LINE		672	612	613				6
108:	LINE		673	613	404				2
109:	LINE		674	402	601				9
110:	LINE		675	603	604				2
111:	LINE		682	605	616				5
112:	LINE		683	616	617				80
113:	LINE		684	617	606				1
114:	SIDE		61	675	64	676			
115:	SIDE		62	61	54	62			
116:	SIDE		63	679	44	680			
117:	SIDE		64	69	670				
118:	SIDE		65	671	681				
119:	SIDE		66	678	66	67			
120:	SIDE		67	682	683	684			
121:	REGION	1	7	-71	71	72	-670		
122:	REGION	1	11	-781	-780	-72	-73		
123:	REGION	1	12	-76	73	-782	-12		
124:	LINE		71	610	707				23
125:	LINE		72	701	702				3
126:	LINE		73	702	703				3
127:	LINE		74	703	104				4
128:	LINE		75	104	103				3
129:	LINE		76	102	704				10
130:	LINE		77	704	705				2
131:	LINE		78	705	706				6
132:	LINE		79	706	611				5
133:	LINE		780	707	701				3
134:	LINE		781	707	703				3
135:	LINE		782	103	706				10
136:	POINT		701	35.5		204.			
137:	POINT		702	32.25		204.			
138:	POINT		703	32.25		201.			
139:	POINT		704	22.5		182.			
140:	POINT		705	23.25		182.			

B-. 1 QMESH Data Deck (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER	12345678901234567890123456789012345678901234567890123456789012345678901234567890								
141:	POINT	706	29.25		182.				
142:	POINT	707	35.5		201.				
143:	SIDE	71	781	74					
144:	SIDE	72	13	782	79				
145:	SIDE	73	77	78					
146:	POINT	801	40.		165.25				
147:	POINT	802	40.25		165.25				
148:	POINT	803	40.25		27.				
149:	POINT	804	40.		27.				
150:	POINT	805	41.		27.				
151:	POINT	806	41.		165.25				
152:	LINE	81	603	805		8			
153:	LINE	82	805	803		2			
154:	LINE	83	803	804		1		108	
155:	LINE	84	804	804		8		109	
156:	LINE	85	805	806		90			
157:	LINE	86	806	802		2			
158:	LINE	87	802	803		90		107	
159:	LINE	88	806	808		8			
160:	LINE	89	607	801		8		105	
161:	LINE	890	801	802		1		106	
162:	POINT	901	22.5		175.				
163:	POINT	902	22.5		173.				
164:	POINT	903	22.5		24.25				
165:	POINT	904	0.		12.5				
166:	POINT	905	0.		10.5				
167:	POINT	906	22.75		10.5				
168:	POINT	907	22.5		14.5				
169:	POINT	908	20.5		12.5				
170:	POINT	909	20.5		14.5				
171:	LINE	91	704	901		8		110	
172:	LINE	92	901	902		2		111	
173:	LINE	93	902	903		80		112	
174:	LINE	94	903	401		2			
175:	LINE	95	305	906		3			

B-1. QMESH Data Deck (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER		1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
176:	LINE		96	906	905		11		
177:	LINE		97	905	904		4		
178:	LINE		98	904	908		10	115	
179:	LINE CIRC		99	908	907	909	3	114	
180:	LINE		910	907	903		10	113	
181:	LINE		911	401	304		16		
182:	LINE		912	613	705		8		
183:	SIDE		91	91	92	93			
184:	SIDE		92	46	673	912			
185:	REGION	1	17	-913	-914	-98	-97		
186:	REGION	1	18	-915	-916	-34	-95		
187:	REGION	1	19	-94	-917	-918	-910		
188:	REGION	1	20	93	-914	-99	-918	-920	
189:	POINT		911	23.25		14.5			
190:	POINT		910	20.5		10.5			
191:	POINT		912	23.25		10.5			
192:	SIDE		93	915	919				
193:	LINE		913	905	910		10		
194:	LINE		914	908	910		4		
195:	LINE		915	906	912		2		
196:	LINE		916	912	304		3		
197:	LINE		917	401	911		10		
198:	LINE		918	911	907		2		
199:	LINE		919	906	910		3		
200:	LINE		920	911	912		7		
201:	REGION	1	21	-77	-921	-922	-923		
202:	REGION	1	22	-922	-673	-925	-926		
203:	REGION	1	23	-925	-46	-94	-929		
204:	LINE		921	705	613		8		
205:	LINE		922	613	901		2		
206:	LINE		923	901	704		8		
207:	LINE		925	404	902		2		
208:	LINE		926	902	901		2		
209:	LINE		929	903	902		80		
210:	SCHEME			SIRS(DVSI)P					
211:	END								

B-1. QMESH Data Deck (cont)

CARD	COLUMNS	1	2	3	4	5	6	7	8
NUMBER		1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890	1234567890
1:	P-L-P	101	15	105					
2:	END								

B-2. RENUM Data Deck

DISTRIBUTION:

U. S. Nuclear Regulatory Commission (16)
Washington, DC 20555

Attn: W. R. Lahs, RES (5)
W. F. Anderson, OSD
D. R. Hopkins, OSD
D. J. Guzy, OSD
C. E. MacDonald, NMSS
C. R. Chappell, NMSS
C. R. Shieh, NMSS
C. R. Marotta, NMSS
W. H. Lake, NMSS
R. H. Odegaarden, NMSS
M. Hartzman, NRR
C. Troutman, ADP Support Div.
Office of Admin.

Energy Res. and Dev. Admin (3)
Washington, DC 20545
Attn: W. A. Brobst, ECT

Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201
Attn: R. A. Robinson

Brookhaven National Laboratory
Upton, LI, NY 11973
Attn: M. Reich

Westinghouse Electric Corp.
Bettis Atomic Power Laboratory
P.O. Box 79
West Mifflin, PA 15122
Attn: Librarian

General Electric Co.
175 Curtner Avenue
San Jose, CA 95125
Attn: R. H. Jones

John Henry Associates
60 Hickory Drive
Waltham, MA 02154

Los Alamos Scientific Laboratory
P.O. Box 1663
Los Alamos, NM 87544
Attn: T. Butler, WX-8

NL Industries, Inc.
919 Market Street
Wilmington, DE 19801
Attn: C. Williams

Nuclear Assurance Corp.
24 Executive Park West
Atlanta, GA 30329
Attn: R. Best

Oak Ridge National Lab. (6)
P.O. Box X
Oak Ridge, TN 37830

Attn: R. D. Seagren
L. B. Shappert
J. H. Evans
G. E. Whitesides
L. Petrie
S. Iskander

Allied General Nuclear Services
P.O. Box 847
Barnwell, SC 29812
Attn: R. W. Peterson

Argonne National Laboratory (3)
9700 South Cass Avenue
Argonne, IL 60439
Attn: J. Natale
M. Butler, Code Center
H. Edwards, Code Center

E. I. du Pont de Nemours & Co.
Savannah River Laboratory
Aiken, SC 29801
Attn: J. W. Langhaar

Knolls Atomic Power Laboratory
P.O. Box 1072
Schenectady, NY 12301

Transnuclaire, Inc.
1 North Broadway
White Plains, NY 10601

General Atomic Co.
P.O. Box 81608
San Diego, CA 92138
Attn: Library

University of California
Lawrence Livermore Laboratory
P.O. Box 808
Livermore, CA 94550
Attn: R. T. Langland

U. S. Energy Res. and Dev. Admin. (2)
Albuquerque Operations Office
P. O. Box 5400
Albuquerque, NM 87115
Attn: R. A. Hagen
E. R. Orner

DISTRIBUTION (cont)

1280 T. B. Lane
1281 S. W. Key
1281 A. F. Schkade (10)
1282 T. G. Priddy
1284 R. T. Othmer
2613 M. R. Scott
2613 R. E. Jones (10)
2614 P. Vandelinder
5430 R. M. Jefferson
5431 C. M. Stone
5431 W. A. Von Reisemann (10)
5431 D. W. Lobitz
5433 J. M. Freedman
5433 G. C. Allen
5711 E. L. Harley
8120 W. E. Alzheimer
8266 E. A. Aas
3141 C. A. Pepmuller (Actg) (5)
3151 W. L. Garner (3)
For ERDA/TIC (Unlimited Release)
ERDA/TIC (25)
(R. P. Campbell, 3172-3)