NUREG/CR-1567
LA-8427-MS

Informal Report

TRAP: Plotting Package for TRAC

University of California

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos, New Mexico 87545

This report was not edited by the Technical Information staff.

NUREG/CR-1567
LA-8427-MS
Informal Report

R4

TRAP: Plotting Package for TRAC

James C. Ferguson

Manuscript submitted: June 1980
Date published: Augu.:t 1980

Prepared for
Division of Reactor Safety Research
Office of Nuclear Regulatory Research
US Nuclear Regulatory Commission
Washington, DC 20555

NRC FIN No. A7016

UNITED STATES
DEPARTMENT OF ENERGY
CONTRACT W-7405-ENG. 36

CONTENTS

TRAP: PLOTTING PACKAGE FOR TRAC

by

James C. Ferguson

ABSTRACT

The TRAP computer program is a versatile postprocessor routine that performs a variety of TRAC-data display tasks. TRAP (TRAC Plot) can graph curves, surfaces, and component structures and can easily be extended to more complex applications. Users communicate with TRAP by a sequence of command language statements allowing a high degree of flexibility in defining their plot requests. Output media for the resulting graphs include 16-mm or 35-mm (black and white or color), 105-mm (fiche) film, and direct imaging on a Tektronix terminal screen.

I. INTRODUCTION

The TRAC (Transient Reactor Analysis Code) computer program[1] outputs nuclear reactor parameters and fluid flow variables, which we call dependent variables, at selected times in the calculation. Because these variables also may change with respect to position within the reactor components, we represent them by either $F(\underline{p},t)$ or $\underline{F}(\underline{p},t)$. Here, F denotes a scalar-valued function (such as liquid density), while $\underline{F}$ denotes a vector-valued function (such as liquid velocity). The symbols $\underline{p}$ and t denote, respectively, spatial and time independent variables, where for example, $\underline{p} = (r, \theta, z)$ represents a point in cylindrical coordinates within a vessel component.

The TRAP graphics program basically produces computer-generated plots of dependent vs independent variables in a variety of useful combinations, including curve (Fig. 1) and surface (Fig. 2) displays. On option, curves can be drawn smoothly (i.e., with continuous slopes). Also, several curves may be drawn on a single frame (Fig. 3). Surface displays erase hidden lines, and the user can vary viewing position (for example, he may do stereo pair plotting -- see Fig. 2).

Users communicate their plot requests to TRAP by employing a simple yet powerful set of TRAP graphics commands. They can either give these statements to TRAP interactively, thereby responding to program queries, or they can run the program in a "batch" mode by supplying the name of a previously prepared file of commands.

Interesting features of TRAP include a MOVIE option, MACRO-component displays (where individual components, or parts thereof, are drawn -- see Fig. 4), EXTENDED FUNCTION capability (where users define specific dependent variables as mathematical and logical combinations of those supplied by TRAC -- see Fig. 5), SUPERPOSITION of auxiliary data (where data in a peripheral file is co-plotted on the current frame for data comparison applications -- see Fig. 6), FORMATTED data file preparation, as well as others detailed below.

Various output media can be used to exhibit TRAP results. A user can choose from among 16-mm, 35-mm (black and white or color), 105-mm (fiche) film, or direct Tektronix tube imaging.

One of the guidelines in developing TRAP has been to construct a program capable of handling basic graphics as well as special plotting applications. For example, it is relatively easy to add a particle-pushing movie program,[2] where certain combinations of the density, velocity, and void fraction dependent variables are used to depict two-phase fluid flow in a reactor vessel. Such extensions provide more realism and drama in the graphics. TRAP is also structured to maximize transportability of the code to other computing environments. Concepts such as top-down design, programming, and internal documentation guided the development of the package. (Details of system dependent constructions are planned for later documentation.)

Expansions on these topics appear subsequently, where emphasis is placed on learning to use TRAP quickly and effectively. Later documents will cover program and data structures as well as coding details, which are of considerable value to those who either (1) need to modify TRAP to fit special

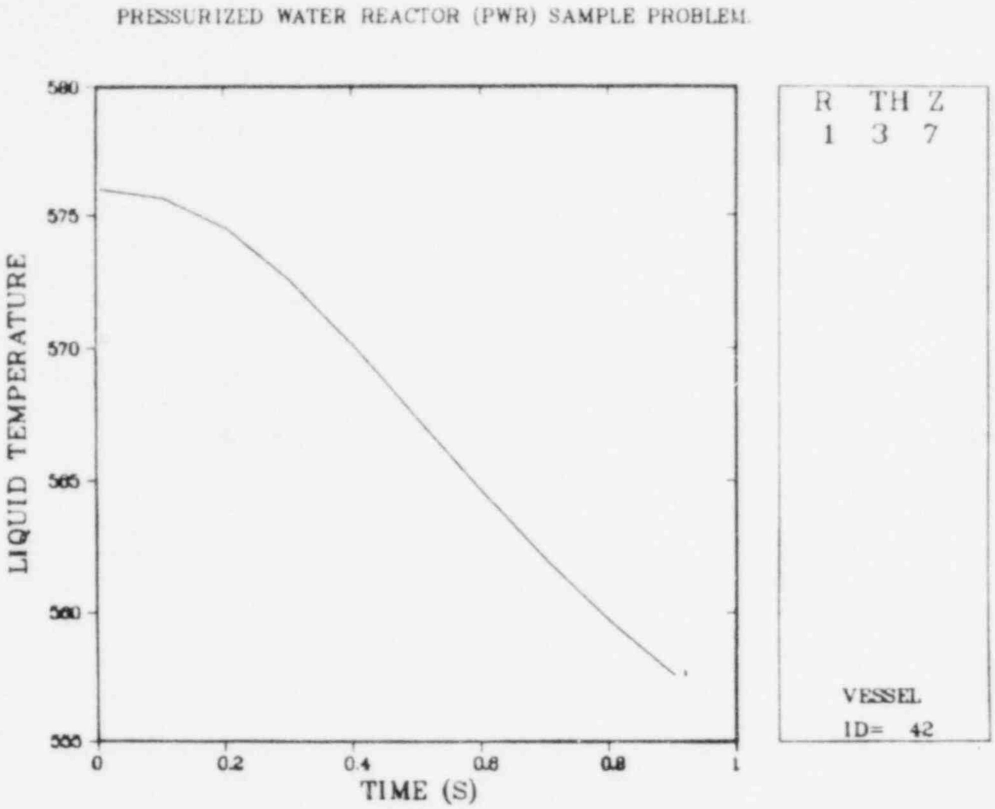


Fig. 1. Liquid temperature vs time.

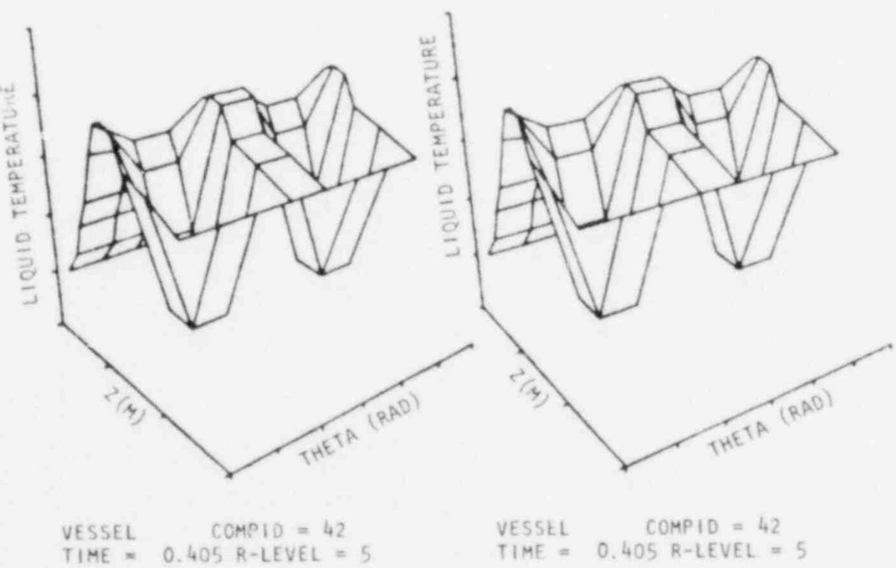


Fig. 2. Example of surface plot.

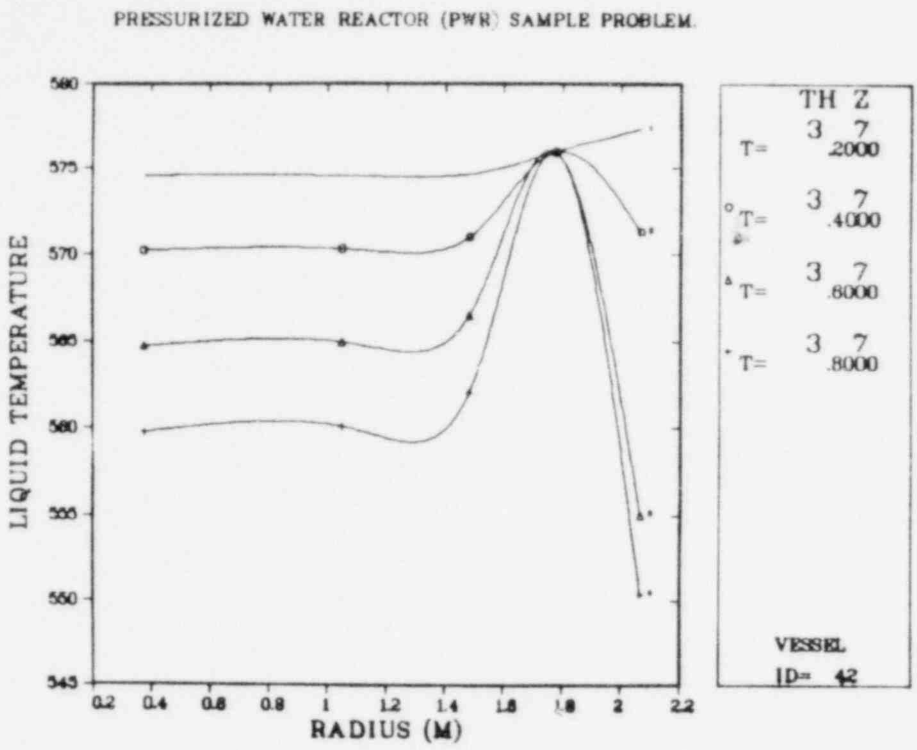


Fig. 3. Liquid temperature vs radius.

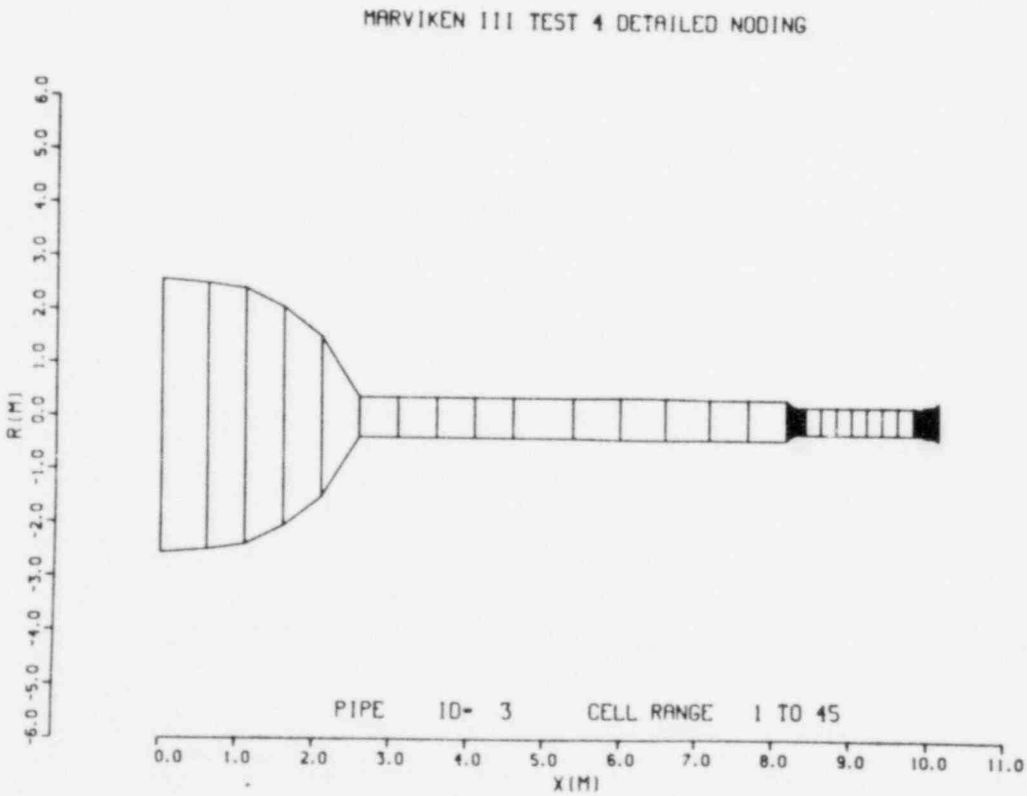


Fig. 4. Pipe component.

PRESSURIZED WATER REACTOR (PWR) SAMPLE PROBLEM

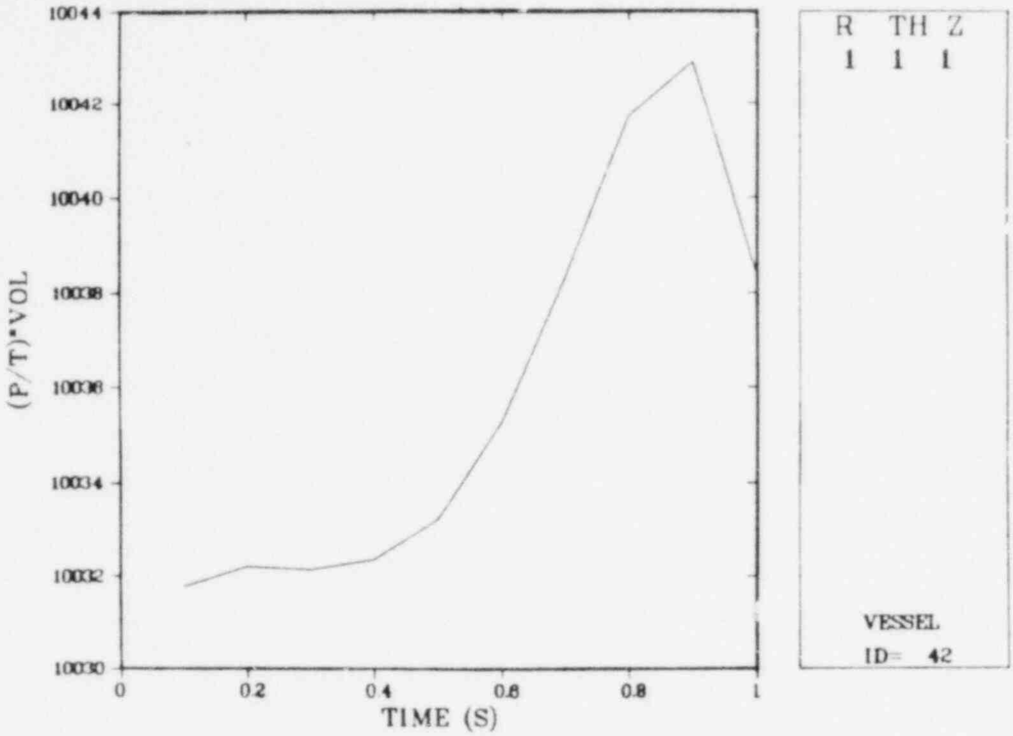


Fig. 5. Extended function example.

PRESSURIZED WATER REACTOR (PWR) SAMPLE PROBLEM

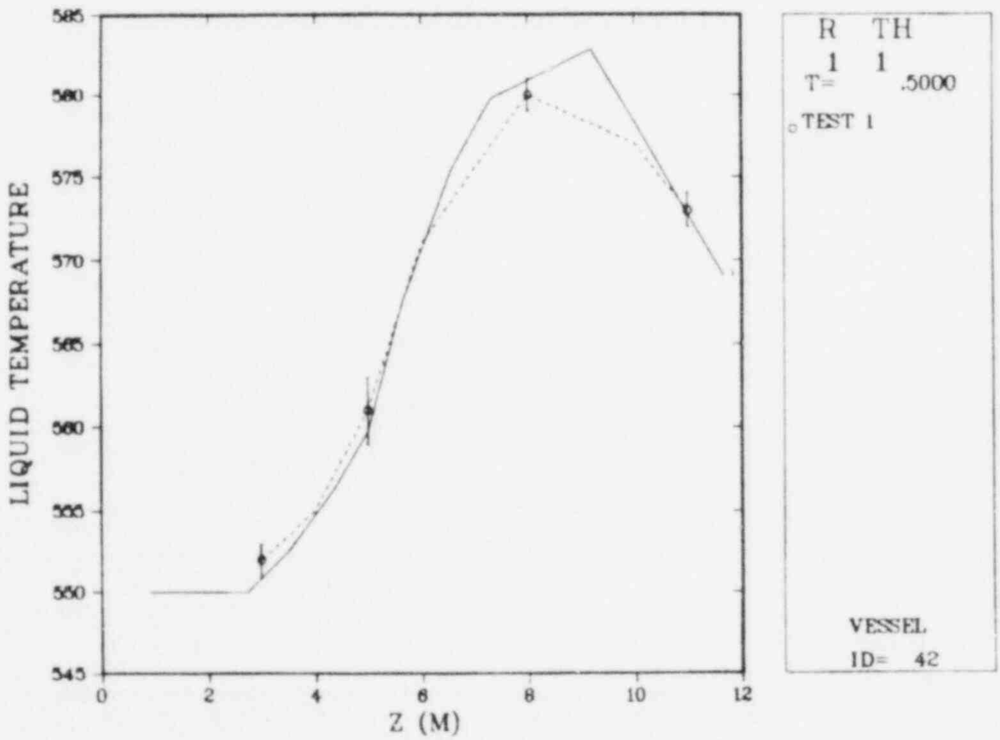


Fig. 6. INCLUDE example.

circumstances, or (2) desire to use TRAP on data provided by programs other than TRAC (in which case, the detailed structure of the TRAP graphics file, COMPij, must be known).

The computing environment is currently CDC 7600 (60-bit word, 10-characters/word), LTSS/FTN with DISSPLA[3] graphics software.

II. BASIC CONCEPTS AND CAPABILITIES

Recall the function representation of dependent variables, F(p,t), mentioned above. Many tasks given to TRAP can be conceptually organized by considering the extent of the domain variable p. For instance, consider the following cases:

(1) p is not defined (for example, component-valued functions such as reactor power), where

POWER = f(t).

(2) p is one-dimensional (for example, a pipe component with geometric coordinate x), where

PRESSURE = f(x,t).

(3) p is two dimensional. For example, a rod with

TEMPERATURE = f(r,z,t)

where

r = node radial coordinate and
z = axial coordinate.

(4) p is three dimensional; for example, a vessel component with

VOID FRACTION = f(r,θ,z,t),

where

θ = azimuthal coordinate,

r = vessel radial coordinate, and

z = vessel axial coordinate.

The current TRAP implementation treats functions that are cell center-valued (e.g., pressure), cell edge-valued (e.g., velocity), component-valued (e.g., reactor power), or rod-valued (e.g., max hot rod temperature).

For one-dimensional components such as pipes, the specific plotting combinations are

Function	Description
f vs t	curve
f vs x	curve
f vs (x,t)	surface
f(x) vs g(x)	correlation curve
f(t) vs g(t)	correlation curve

Here, we specify only the free independent and the dependent variables. The fixed independent variable is not explicitly named as it is easily inferred. Independent variables are either fixed or free, where free indicates the variable that traverses a curve or surface; that is, it appears as an abscissa coordinate. The fixed independent variable serves as a general index in a set of plotted curves. For each such fixed value, a separate curve is generated. The curves may be members of a family all appearing on one frame; or, the individual curves may be plotted on distinct frames. For example, plot y = f (x,t) where t is free and x is fixed at the first through fifth cells of the component. This produces five separate curves, one for each cell in the range one through five. The abscissa variable for each of these curves is time.

Correlation curves are actually parametric curves with the free variable acting as the defining parameter for the points. These curves are not necessarily one-to-one off either axis, and in fact may self-intersect. An example is to cross plot vapor temperature (TV) vs liquid temperature (TL) over the time range 0.0 to 1.0 at intervals of 0.2 s. Here, six values of TL

$(TL_1,\ldots,TL_6)$ define the abscissa coordinates in a two-dimensional plot while six values of TV $(TV_1,\ldots,TV_6)$ define the ordinate coordinates. The pairs (TL_i,TV_i) define points on the correlation curve, where (TL_i, TV_i) corresponds to the time value = 0.2*(i-1), for i = 1,...,6.

For three-dimensional components the following combinations are available.

Function	Description
f vs r	curve
f vs θ	curve
f vs z	curve
f vs t	curve
f vs (r,z)	surface
f vs (r,θ)	surface
f vs (θ,z)	surface
f(r) vs g(r)	correlation curve
f(θ) vs g(θ)	correlation curve
f(z) vs g(z)	correlation curve
f(t) vs g(t)	correlation curve

The surface displays are three-dimensional perspective views of the surface, where hidden lines are eliminated. Future variations will involve contouring these surfaces and either (1) displaying the level curves as they appear on the surface in three dimensions or (2) displaying their projections in the plane of the function domain.

III. USING TRAP

TRAC produces a graphics file, TRCGRF, which is converted into a set of reorganized component data files by program EXCON. These data files are named COMPij where ij is an integer corresponding to the user-defined component number. Program TRAP employs the COMPij files as a data base for its work in that it extracts appropriate geometry and physical variables from them in preparation for producing plots. (See Fig. 7 for a compact illustration of overall data flow between controllees.)

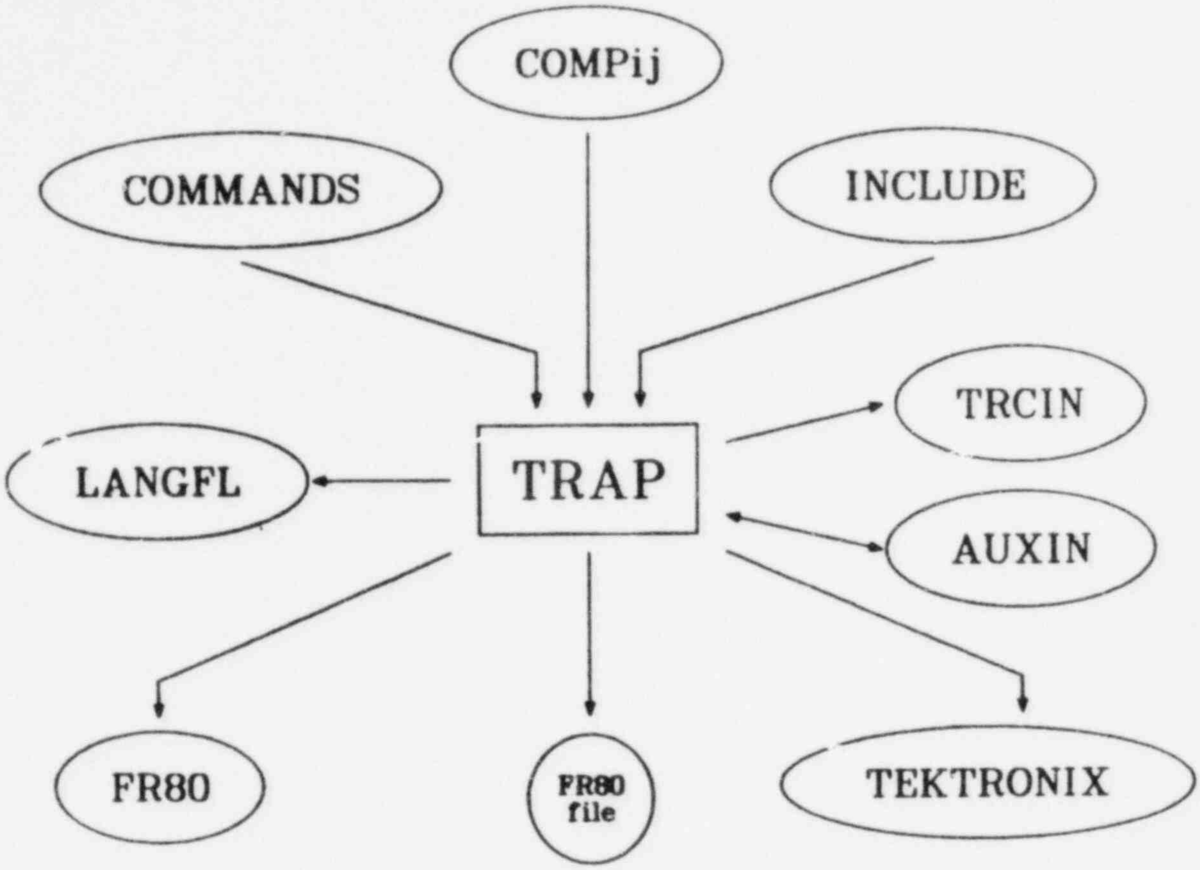


Fig. 7. TRAP program data flow.

The user communicates with TRAP either in a query mode or by a prepared file of plot commands. In either mode, a user gives the program a set of language statements forming a plot request. A language statement, or command, consists of an ordered set of space delimited arguments, such as

FIXED RADIUS 1 3

The first argument is a command primary keyword. It may be followed in the string by other secondary entries that define arguments to the command. Here, the keyword FIXED tells TRAP that a fixed independent variable is being defined. Also, the phrase

RADIUS 1 3

tells TRAP the independent variable is the vessel radius that ranges from the first through the third radial cell. At the Los Alamos Scientific Laboratory, several commands may be entered on one line if they are separated by an "escape" control character. Command syntax varies according to command type. A list of available TRAP commands is given in Sec. IV, where the specific format of each is defined.

The command entries are used by the program to assemble a definition of the plot task for dependent and independent variable specifications, component ID, etc. The order of the language statements is flexible, as the program accumulates a plot request (or plot problem definition) by analyzing entries independently. When the user is satisfied with the definition, the DISPLAY command is entered to terminate the plot request. TRAP then checks the current user-supplied set of commands for completeness and consistency, and if satisfied, TRAP extracts pertinent data from the component files and develops the necessary plot frames.

If the commands are given in a file, then they appear exactly as they would in an interactive terminal session, except for the prompt symbol μ.

Example.

Plot liquid temperature vs cell position for component 16. Here, this component is a pressurizer and is a one-dimensional model. First generate the component data file, COMP16, by running EXCON[4] (i.e., upper-case entries are user-supplied),

```
EXCON GO / 1 1.5

Enter Space Delimited Component Numbers
? 16
All Done
```

Here, TRCGRF is the graphics file produced by program TRAC. Next run TRAP,

```
TRAP / 1 1.5
Enter general Title
TEST CASE
```

```
Enter Problem Specification
? COMPID 16
? FIX TIME .4
? FREE X 1 5
? DEP TL
? DISPLAY
Plot Request Completed
? END
All done
```

This produces the plot shown in Fig. 8. We define the component ID number as 16; set TIME as the fixed independent variable at 0.4 s, set the cell position coordinate, X, as the free independent variable ranging from the first through the fifth cell; and define TL (liquid temperature) as the dependent variable.

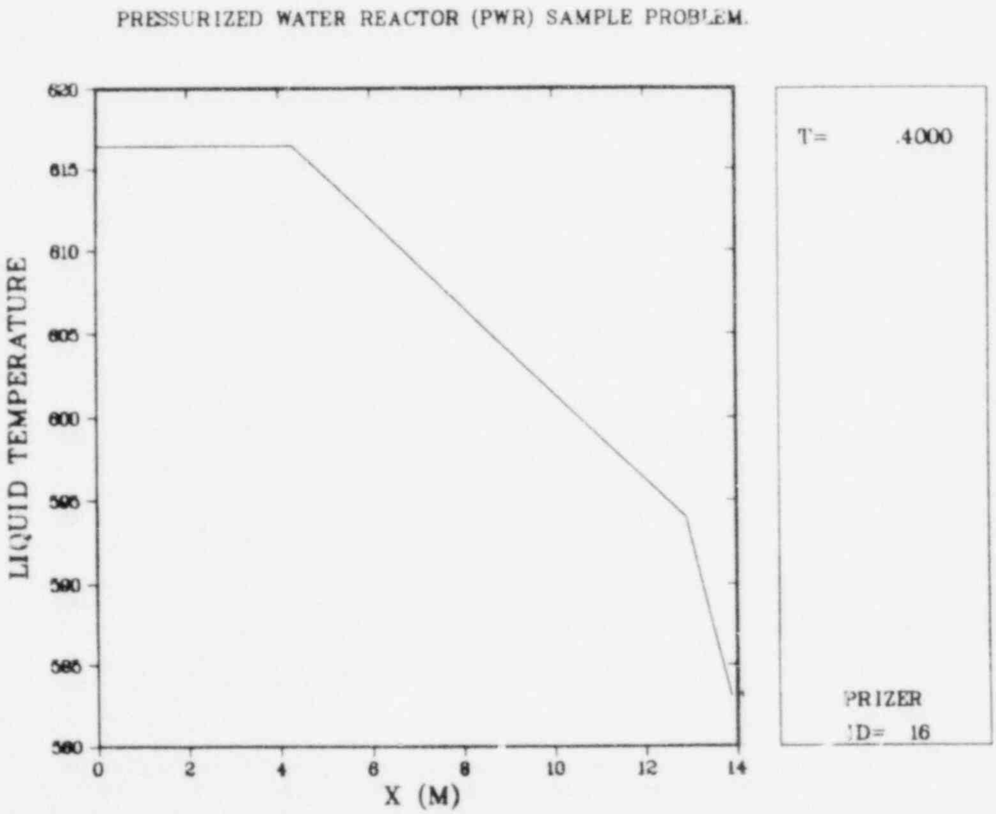


Fig. 8. Liquid temperature vs X.

Example.

Plot, for a three-dimensional component, a curve relating liquid temperature and axial position (the free independent variable). Let the component ID be 42 and assume the associated data on file COMP42 resides in the user's local file space. Let the values of the fixed independent variables be

1st radial cell,
1st angular cell,
t = .5 s.

Also, for this example, superimpose data with error bars from an auxiliary data file. The information in this file is given in Table 1.

TABLE I
AUXFIL DATA

1 1			
2 7 2			
3 TEST 1			
4	3	552	1
5	4	555	1
6	5	561	2
7	6	571	2
8	8	580	1
9	10	577	1
10	11	573	1

The interactive session is

TRAP / 1 1.5
Enter General Title
TEST CASE
Enter Problem Specifications

```
? COMPID 42
? FIXED RADIUS 1
? FIXED THETA 1
? FIXED TIME .5
? FREE Z 1 11
? INCLUDE AUXFIL
? DISPLAY
Plot Request Completed
? END
All Done
```

The user sets the component identification (ID) at 42 (a vessel), fixes RADIUS and THETA as two of the independent variables (that is, he fixes both radius and theta at the first cell), fixes TIME at 0.5 s, defines the free independent variable as Z (ranging from the first z-level through the eleventh z-level), defines the dependent variable mnemonic as TL (which is equivalent to liquid temperature -- these mnemonics are generated by TRAC and saved in the TRCGRF output file), and finally asks TRAP to co-plot data found in the auxiliary file, AUXFIL. He finishes his plot problem definition with the DISPLAY command and terminates the program with the END statement. (The output plot is given in Fig. 6.)

For this example, if the user chooses not to be completely interactive, he prepares a file containing

```
TEST CASE
COMPID 42
FIXED RADIUS 1
FIXED THETA 1
FIXED TIME .5
FREE Z 1 11
DEP TL
INCLUDE AUXFIL
DISPLAY
END
```

If we name this file TRAPIN, the terminal session is

```
TRAP FILE TRAPIN / 1 1.5
Plot Request Completed
All Done
```

Another Example.

This is the same as the first example, except we wish to additionally plot a TL vs X curve for TIME = 0.7 s. The session is

```
TRAP / 1 1.5
Enter General Title
TEST CASE
Enter Problem Specification
? COMPID 16
? FIX TIME .4
? FREE X 1 5
? DEP TL
? DISPLAY
Plot Request Completed
? FIX TIME .7
? DISPLAY
Plot Request Completed
? END
All Done
```

Here, we demonstrate the property of "file memory." TRAP remembers the user commands from one plot request to another, and in this example all previous entries are retained except for the one replaced.

Execution line arguments (order-independent) may be specified in addition to FILE FILENAME. These include device or film type, release codes, an execution time flag, plot title option, and two formatted output file options. The film choices are

35-mm black and white slides,
35-mm color slides,
16-mm movie,
16-mm color movie, or
105-mm microfiche.

To enable these options, enter 35, 35c, 16, 16c, or 105; or specify TEK on the execution line if direct Tektronix screen output is required. If you specify KEEP, then the graphics routines define a plot file that uses the SCAN utility[5] upon termination of TRAP. This file may later be released to the system for film production by employing the CGSGIVE utility[6]

CGSGIVE I=FR80t / t v

If you specify GIVE on the TRAP execution line, then the film file is automatically released, and you will not be able to view the frames with SCAN. If you want intermediate execution-time information, then type TIME on the execution line. If you enter TITLE on the execution line, then TRAP uses the title you supply in the command language file as a plot title for each frame.

Two keywords, TRACIN and AUXFILE, also can appear in the execute line options. Each of these keywords causes TRAP to create a formatted output file (named TRCIN or AUXIN, respectively) instead of graphical output. When the keyword TRACIN is specified, the data requested by the user is written to the TRCIN file in the TRAC input load format. Keyword AUXFILE causes the data to be written in the TRAP INCLUDE input format for plotting as data points. The two keywords are mutually exclusive and TRAP will act on the last one specified if both are accidentally included. Use of these keywords limits the specification of data to a single independent and a single dependent variable, as well as to single values of all fixed variables. The TRACIN keyword produces both independent and dependent variable values if the independent variable is time, but only dependent variable values if the independent variable is spatial. Thus, the output may be included in a TRACIN file directly as either boundary conditions (by a time-dependent FILL or BREAK) or as initial conditions for a renoded component. The AUXFILE keyword always produces both independent and dependent variable values, with corresponding half-widths set to zero.

The defaults for the execution line arguments are

(a) language statements entered in query mode,
(b) 105 mm fiche output,
(c) KEEP release code,
(d) no intermediate timing printout, and
(e) use of the title from the COMPij file as a plot title.

An example of the use of execution line arguments follows.

```
TRAP 35C FILE INFILE GIVE TITLE / 1 1.5
Enter General Title
TEST CASE
    .
    .
    .
   etc.
```

Here,

(1) each frame has TEST CASE as a main title,
(2) 35-mm color slides are generated,
(3) TRAP processes a previously prepared file of commands,
(4) the film file produced by TRAP is automatically given to the system.

As TRAP processes a set of command statements, it writes a peripheral file, named LANGFL, which contains every user entry. If the user submits commands one per line, then TRAP does not write given commands that are in error. If the user submits commands multiply per line, then any command in the line that is in error causes TRAP to write none of the commands in this line to LANGFL.

This provides the opportunity to edit a command file created in the query mode and reuse it directly. However, if a user wishes to reuse LANGFL, the name must be changed.

IV. SYLLABUS OF TRAP COMMANDS

The following is a complete list of currently implemented TRAP commands. The commands or independent variable names or secondary keywords can be abbreviated to the first two letters in their names except for COMPID (this is indicated by underlining).

CLEAR names

Erases variables and keywords from program memory; "names" is a space-delimited list of variable or keyword names. If the word "all" is given, then the program re-initializes all variable and keyword names. Example,

CLEAR RADIUS TIME MOVE

COMPID integer

Defines component number from which user wishes to produce plots; example,

COMPID 42

DEP names

Defines dependent variable names; "names" is a space-delimited list of names corresponding to those given in the graphics file, TRCGRF, produced by program TRAC. No more than two may be supplied, unless EXTEND is is force; in which case five may be given. If more than one is defined, then TRAP produces correlation plots. Each use of DEP erases those of previous definitions. Example,

DEP ALPHA P

DISPLAY

Separates plot requests; it tells TRAP to produce plots according to the resident plot problem definition.

DVNAMES

Produces a list of dependent variable names and their expanded definitions for the current component. These are the names allowed in the DEP command. (DVNAMES must not occur before COMPID.)

END

Terminates the program

EXTEND number

Extended function command; user supplies subroutine EXTFNC that transforms a set of dependent variables to another; see Sec. V for details. (EXTEND must precede DEP.)

FIXED name numbers

Defines fixed independent variables; the keyword is followed first, by a secondary keyword naming the independent variable (either X, TIME, RADIUS, THETA, or Z); this is followed next by a space-delimited list of numbers defining the variable range; if TIME, then such numbers as a, b, c, are in floating point mode where a=initial time, b=final time, and c=time increment; if not TIME, then such numbers as i, j, are integers where i=initial cell index and j=final index. Example,

```
FIXED TIME .123 23.78 .017
FIXED THETA 3 7
```

FREE name numbers	Defines free independent variables; it is structured identically to FIXED.
INCLUDE file name	Tells TRAP to superimpose data from an auxiliary file on the current frame. More details are given subsequently in Sec. VI. (The INCLUDE command must be respecified between plot requests.)
LEGEND word1 word2	Replace all TRAP legend box annotations by user-supplied description defined by two 10-character, space-delimited words.
MACRO number	Draw individual component partially or completely. This is currently limited to one-dimensional components. Partial component display results by specifying the cell range in a FREE X command. "Number" is the component ID number.
MOVIE	Enables the movie mode. TRAP presently assumes TIME is the independent variable that indexes the frames.
NOADV	Tells the program not to advance frames during the current plot request. However, TRAP always advances between plot requests.
NPLOT	Suppress plotting of curve and legend references defined in plot request. INCLUDE auxiliary curves are not suppressed.
PORT a b c d	Allows plotting in constrained areas of the current frame. The numbers a, b, c, d are floating point and field free, where a=lower x-value of window, b=upper x-value, c=lower

y-value, d=upper y-value. The numbers a, b, c, d are each in the unit interval from 0. to 1. For example,

PORT 0. .5 .5 1.

causes the plot to be displayed in the north west quadrant of the frame. (See Figs. 11a and 11b.)

ROD m	Dependent variable has rod data structure; m is an integer defining the rod number. ROD is specified only when working with rod dependent variables that are functions of both Z and RADIUS. Rod levels are specified relative to the bottom of the vessel. ROD must occur before RADIUS, Z definitions. The FIXED and FREE commands for Z have alternate formats when ROD is specified, namely, FIXED Z A B n where A = initial z-level (in meters), B = final z-level (in meters), and n = number of z-levels. Similarly for FREE Z A B n. However, only one radial node can be processed in any one plot request.
SCALE name numbers	Absolute scaling option for dependent variables. "Name" is the dependent variable name. "Numbers" are two floating point numbers such as a, b, where a=lower limit,

and b=upper limit. (SCALE must not precede DEP.)

SIDE — Dependent variable represents side tube data. This applies to TEE and STEAM GENERATOR components only.

SMOOTH — Displayed curves are normally piecewise linear through the given set of plotted points. This command interpolates a smooth curve through the points having continuous slope. The curve is, specifically, a parametric cubic spline developed originally by Ferguson[7] and implemented by ISSCO for DISSPLA.[3]

SUMMARY — Defines, by printing to the terminal, the current state of a plot request, that is, the keywords and variable names in TRAP memory.

VIEW a b c — Defines the eye point for a perspective view of a surface. Numbers a, b, c are floating-point and space delimited, where a= x-coordinate of eye point, b=y-coordinate, and c=z-coordinate.

V. EXTENDED FUNCTION CAPABILITY

A user may be interested in plotting dependent variables that are combinations of the variables on a COMPij file. He wishes to transform a given set of dependent variable functions (by a user-supplied subroutine, EXTFNC) to another set

$$\text{EXTFNC: } \{D_1,\ldots,D_k\} \longrightarrow \{E_1,\ldots,E_m\}$$

where the domains for the functions D_i, E_j are given in the plot request.

For example, consider a vessel and the following set of TRAP commands (EXTEND is a flag for the extended function option)

```
COM 42
FI TH 1
FI RA 1
FI Z 1
FR TI 0. 1. .1
EXTEND
DEP ALPHA P TL
DI
EN
```

In this example, the actual dependent variable appearing in the plot might be

$$NEWVAR = ALPHA^2 * P / TL * R_1$$

and the plot would involve NEWVAR vs TIME at the (1,1,1)th R, θ, Z cell. The function domains are all identical as specified in the plot request. Additionally, the user needs geometric information because the first radial width, R_1, occurs in the equation above.

The command language statements that enable the extended function option are

EXTEND n

DEP D_1 D_2 ... D_k ,

where each D_i is a dependent variable name and is defined over the same spatial and time domains given in the plot request. "n" is a code available to EXTFNC as an internal switch for equation selection. Subroutine EXTFNC is a subroutine that transforms a given set of dependent variables to another by special mathematical and logical relations of interest to the user. It is written by him and recompiled into TRAP. EXTFNC is invoked by TRAP repeatedly each time required.

The calling sequence for EXTFNC is:

```
CALL EXTFNC (LENX, DX, LENZ, DZ, LENR, DR,
     LENTH, DTH, VOL, FA, MDEP,
     DEPIN1, DEPIN2, DEPIN3, DEPIN4, DEPIN5,
     DPOUT1, DPOUT2, LABL1, LABL2, NAM1, NAM2, ICODE)   ,
```

where

LENX	--	length of Δx array given in plot request
DX	--	array of Δx values, length LENX
LENZ	--	length of Δz array given in plot request
DZ	--	array of Δz values, length LENZ
LENR	--	length of Δr array given in plot request
DR	--	array of Δr values, length LENR
LENTH	--	length of $\Delta\theta$ array given in plot request
DTH	--	array of $\Delta\theta$ values, length LENTH
VOL	--	array of volumes, 1D or 3D component
FA	--	array of flow areas, 1D(LENX+1) or 3D component
MDEP	--	number of incoming dependent variables on input; number of outgoing dependent variables on output
DEPIN1	--	first input dependent variable array
DPOUT1	--	first dependent variable output array
LABL1	--	axis label for DPOUT1, 2 words; output
NAM1	--	DPOUT1 mnemonic; output
ICODE	--	input flag set by user in EXTEND command, which is available for use by subroutine EXTFNC

Five incoming dependent variables can be used along with any or all of the available DEPIN arrays, depending on how the DEP command is entered in the plot request. Also, two dependent variable output arrays exist.

The structure of the dependent variable data arrays depends on component type and on the nature of the plot request, specifically the independent variables definition. If the component is one-dimensional, then the values in the dependent variable arrays are ordered for increasing cell number. If the component is three-dimensional, then the dependent variable values are ordered first for increasing θ-cell value, then for increasing R-cell value, and finally for increasing Z-cell value. The dependent variable values in the DEPIN arrays exist only over the spatial independent variable ranges, as given in the plot request.

For example, the component is a pipe, and the plot request is

```
TITLE
COM 7
FI X 4 6
FR TI 0. 1. .1
EX
DE TL
DI
EN
```

The dependent variable data, stored in DEPIN1, is ordered as follows

DEPIN1(1) -- TL at 4th cell
DEPIN1(2) -- TL at 5th cell
DEPIN1(3) -- TL at 6th cell

For example, the component is a vessel. The plot request consists of the following command statements

```
TITLE
COM 42
FI RA 3 4
FI TH 1 7
FI Z 4 5
FR TI 0. 1. .1
```

EX
DE TL
DI
EN

The dependent variable data, stored in DEPIN1, is ordered as follows

DEPIN1(1) -- TL at θ, R, Z cell (1, 3, 4)

.

.

.

DEPIN1(7) -- TL at θ, R, Z cell (7, 3, 4)
DEPINL(8) -- TL at θ, R, Z cell (1, 4, 4)

.

.

.

DEPIN1(14) -- TL at θ, R, Z cell (7, 4, 4)
DEPIN1(15) -- TL at θ, R, Z cell (1, 3, 5)

.

.

.

DEPIN1(21) -- TL at θ, R, Z cell (7, 3, 5)
DEPIN1(22) -- TL at θ, R, Z cell (1, 4, 5)

.

.

.

DEPIN1(28) -- TL at θ, R, Z cell (7, 4, 5) .

TRAP performs all LCM/SCM data transfers and manages the time edit blocks as they are given to EXTFNC. TRAP enacts the EXTFNC subroutine after it interpolates dependent variable data at user-supplied time values. TRAP also checks for full LCM/SCM buffers as well as for the movie option. This minimizes the programming involvement for the user. An example of the subroutine EXTFNC follows. Let

$D = (D_1/D_2)*V,$

where

D_1 = first dependent variable,
D_2 = second dependent variable,
V = cell volume, and
D = transformed dependent variable.

Specifically, we are considering a vessel and are interested in the case,

D_1 = pressure
D_2 = liquid temperature.

Our command set to TRAP is

```
COM 42
FI RA 1
FI TH 1
FI Z 1
FR TI 0. 1.
EX
DE P TL
DI
EN
```

This produces a plot (P/TL)*VOL vs TIME at the 1,1,1-th vessel cell. (See Fig. 5 for the resulting plot, where component 42 is a vessel from the USPWR1 problem.) The EXTFNC listing for this simple example is given in Table II.

TABLE II

LISTING OF EXTFNC

```
      SUBROUTINE EXTFNC (LENX, DX, LENZ, DZ, LENR, DR,
     $ LENTH, DTH, VOL, FA, MDEP,
     $ DEPIN1, DEPIN2, DEPIN3,
     $ DEPIN4, DEPIN5, DEPOUT1, DPOUT2, LABL1, LABL2,
     $ NAM1, NAM2, ICODE)
C
C     TRANSFORMS TWO DEPENDENT VARIABLES:
C     DEPNEW  =  DEPIN1/DEPIN2 * VOL
C
C     USED IN CONTEXT OF EXTENDED FUNCTION CAPABILITY
C
C
C     THIS IS TEST VERSION ONLY
C
C     J. FERGUSON   08-21-79
C
C
      REAL DX(1), DZ(1), DR(1), DTH(1), FA(1),
     $ DEPIN1 (LENTH,LENR,LENZ), DEPIN2 (LENTH,LENR,LENZ),
     $ DEPIN3 (LENTH,LENR,LENZ), DEPIN4 (LENTH,LENR,LENZ),
     $ DEPIN5 (LENTH,LENR,LENZ), DPOUT1 (LENTH,LENR,LENZ),
     $ DPOUT2 (LENTH,LENR,LENZ), VOL (LENTH,LENR,LENZ)
      INTEGER LABL1(2), LABL2(2)
C***TRANSFORM VARIABLES
      DO 10 K=1, LENZ
      DO 10 J=1, LENR
      DO 10 I=1, LENTH
      DPOUT1(I,J,K) = DEPIN1(I,J,K)/DEPIN2(I,J,K)*VOL(I,J,K)
   10 CONTINUE
C***DEFINE LABELS & MNEMONICS
      LABL1(1) = 9H(P/T)*VOL
      LABL1(2) = 1H
      NAM1 = 7HP/T*VOL
      MDEP = 1
      RETURN
      END
```

VI. SUPERPOSITION OF AUXILIARY DATA

The INCLUDE operation co-plots data in a peripheral file with data supplied by the TRAC computer program. The user is responsible for consistency and scaling limits.

The format for the auxiliary file is

L

M N

WORD

X_1 Y_1 W_1

.

.

.

X_m Y_m W_m

.

.

. ,

where the entries per line are defined as follows.

L = number of curves in the file.

M = number of points in the curve.

N = point skip integer; the program plots error bars at every N-th point.

(note: L, M, N are entered as space-delimited, field-free integers.)

WORD = curve descriptor (2a10).

(X,Y) = rectangular coordinates of error bar centers.

W = half width of error bar.

[(X, Y, W are given in format (3e10.3)].

See Fig. 6 for an example.

VII. ADDITIONAL EXAMPLES

Most of the following examples are based on a standard problem, USPWR1, which models a reactor system having 42 components.

Component no. 42 is a VESSEL, where

NASX = no. axial cells = 11,

NRSX = no. radial cells = 5,

NTSX = no. angular cells = 8.

Component no 6 is a TEE, where

NCELL1 = no. cells in primary tube = 2,

NCELL2 = no. cells in side tube = 6.

Component no. 16 is a PRESSURIZER having five cells.

(a) Create COMPij files corresponding to components 6, 16, 42:

```
EXCON GO
Enter Space Delimited Component Numbers
  6 16 42
All Done
```

This reformats file TRCGRF (assume it exists as a local file) to files COMP06, COMP16, COMP42.

(b) Create two plot frames during the same session and display them on the Tektronix screen (Figs. 1, 3):

```
TRAP FILE TRAPIN TEK / t v
Plot Request Completed
Plot Request Completed
All Done
```

where TRAPIN contains

```
TEST CASES
COM 42
FI RA 1
FI TH 3
FI Z 7
```

```
FR TI .005 1. .1
DE TL
SCALE TL 545. 580.
DI
CL RA TI
FI TI .2 .8 .2
FR RA 1 5
SM
NO
DI
EN
```

The first plot graphs LIQUID TEMPERATURE as a function of TIME at the r=1, θ=3, z=7 cell in the vessel. The second graph plots LIQUID TEMPERATURE as a function of RADIUS at TIME = 0.2 to 0.8 at 0.2 s over vessel cells with θ=3, z=7, (cell indices). This second plot contains four curves co-plotted on the same frame because the NOADV option is set.

(c) Create a movie of PRESSURE vs RADIUS:

```
TRAP FILE FILE1 16 GIVE / t v
Plot Request Completed
All Done
```

where FILE1 contains

```
TEST CASE
COM 42
FI TH 7
FI Z 2
FI TI .001 1. .005
FR RA 1 5
DE P
MO
DI
EN
```

This produces a short movie of about 200 frames, lasting about 12 s when projected at 16 frames/s. The film files are atuomatically released to the FR80 plotter.

(d) Create two plots. The first is LIQUID HEAT TRANSFER COEFFICIENT vs TIME for the side tube of the TEE, and the second is LIQUID TEMPERATURE vs X for the PRESSURIZER.

```
TRAP FILE INFILE 35C GIVE / t v
Plot Request Completed
Plot Request Completed
All Done
```

where INFILE contains

```
TEST CASES
COM 6
FI X 4
FR TI .005 1. .05
DE HLSID
SI
DI
CL ALL
COM 16
FI TI .4
FR X 1 5
DE TL
DI
EN
```

The first plot request produces one 35-mm color frame (b and w representation in Fig. 9). The second plot request also produces one 35-mm color frame (b and w in Fig. 8). Both are automatically released to the system.

PRESSURIZED WATER REACTOR (PWR) SAMPLE PROBLEM

TYPICAL US PWR WITH FOUR LOOPS---STEADY-STATE CALCULATION

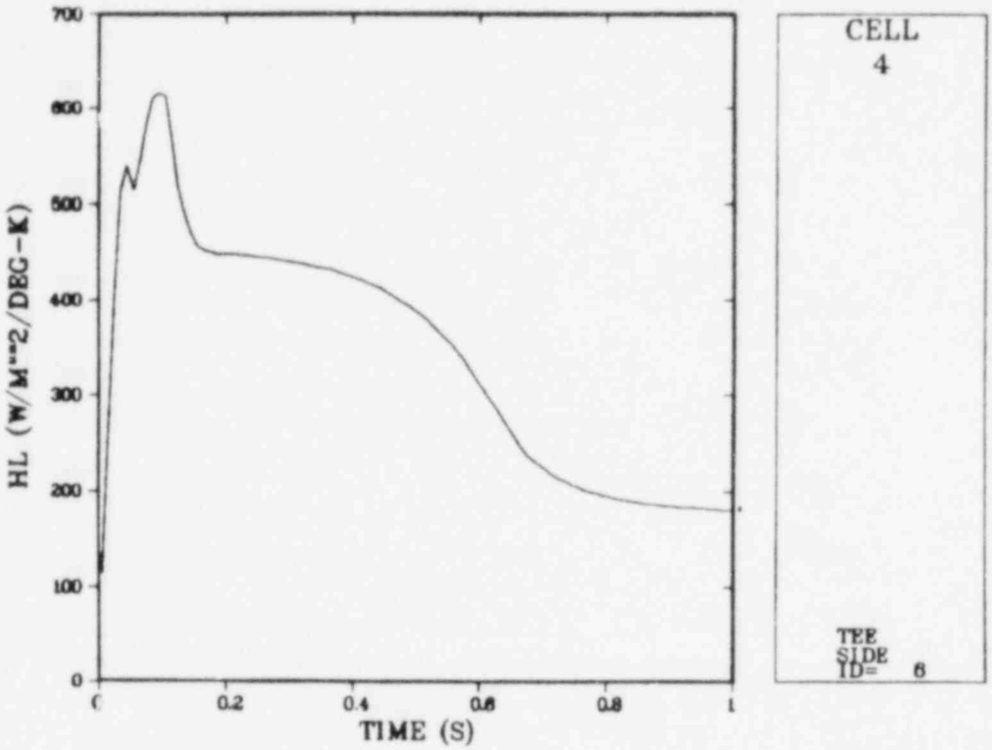


Fig. 9. HLSID vs TIME.

(e) In this example, we temporarily deal with another reactor system, display component 16 (See Fig. 4), which happens to be a pipe component having 45 cells:

```
TRAP TEK / 1 1.5
Enter General Title
TEST CASE
Enter Problem Specifications
? MA 16
? DI
Plot Request Completed
? EN
All Done
```

(f) Plot average rod temperature vs Z for rod number 13 at radial node number 8 (See Fig. 10):

```
TRAP TEK / 1 1.5
Enter General Title
TEST CASE
Enter Problem Specifications
? COM 42
? ROD 13
? FI RA 8
? FR Z 0. 12. 12
? FI TI .2
? DE AVRODTMP
? DI
Plot Request Completed
? EN
All Done
```

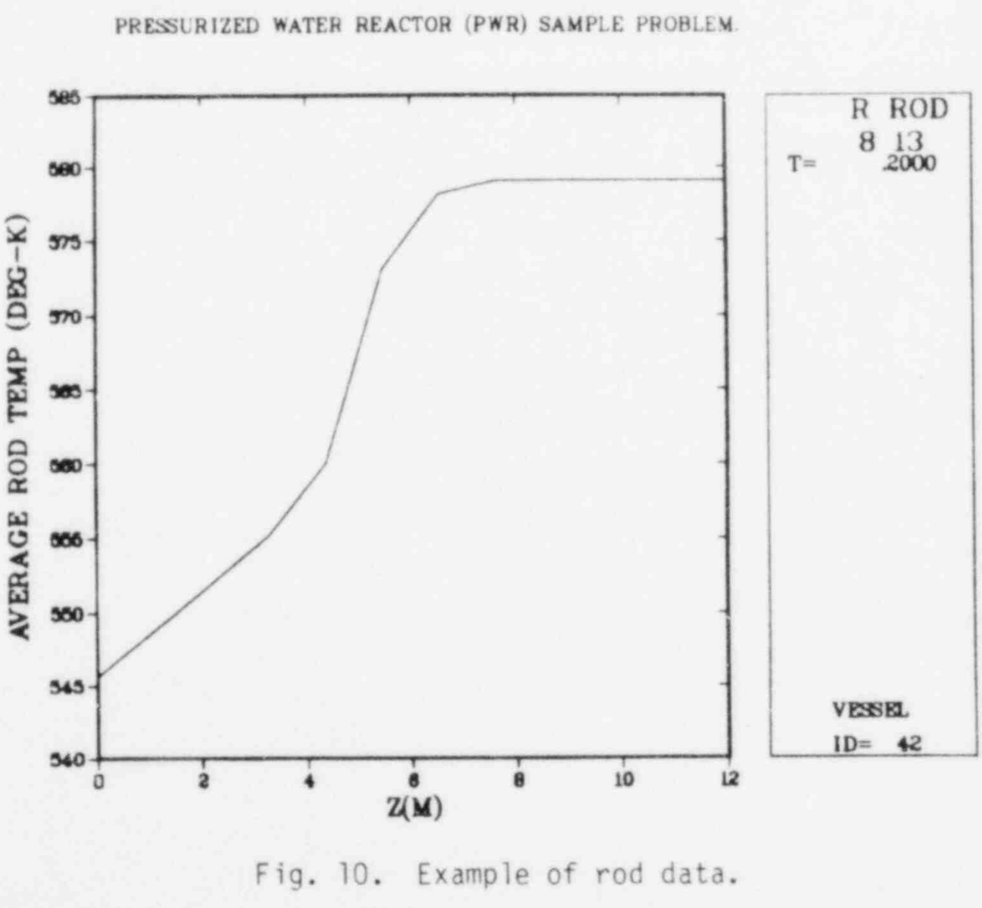


Fig. 10. Example of rod data.

(g) Example of a surface plot -- display liquid temperature as a function of radius and theta for the vessel component number 42:

```
TRAP FILE INTRAP / 1 1.5
Plot Request Completed
All Done
```

where file INTRAP contains

```
TEST CASE
COM 42
FI RA 5
FI TI .405
FR TH 1 8
FR Z 1 11
DE TL
VIEW 11. 30. 600.
DI
VIEW 12. 29. 600.
DI
EN
```

This generates a stereo pair, with the first frame corresponding to a right eye image. The two frames are shown together in Fig. 2.

(h) Example of use of DVNAMES and SUMMARY commands:

```
TRAP / 1 1.5
Enter General Title
TEST CASE
Enter Problem Specifications
? COM 16
? DV
```

DEPENDENT VARIABLE DICTIONARY

COMP = 16 TYPE = PRIZER NCELLS = 5

MFLOW	MASS FLOW-RATE (kg/s)
RHOM	MIXTURE DENSITY (kg/m**3)
TSAT	SATURATION TEMP (deg-k)
CFZ	FRICTION FACTOR
ALPHA	VOID FRACTION
VM	MIXTURE VELOCITY (m/s)
VR	RELATIVE VELOCITY (m/s)
TL	LIQUID TEMP (deg/k)
TV	VAPOR TEMP (deg-k)
P	PRESSURE (pa)
VFLOW	DSCHRG VOL FLOW (M**3/s)
VLOSS	LIQD VOL DSCHRGD (M**3)
HEIGHT	WATER LEVEL (m)
POWER	HEATER/SPRAYER POWER (w)

```
? FI X 3
? FR TI 0. 1. .1
? DE TL
? SU
```

SUMMARY OF USER SPECIFICATIONS

COMPONENT ID = 16

PRIMARY KEYWORD OPTIONS ARE -

COMPID

DEP

FIXED

FREE

SUMMARY

DVNAMES

FREE INDEPENDENT VARIABLES ARE -

TIME

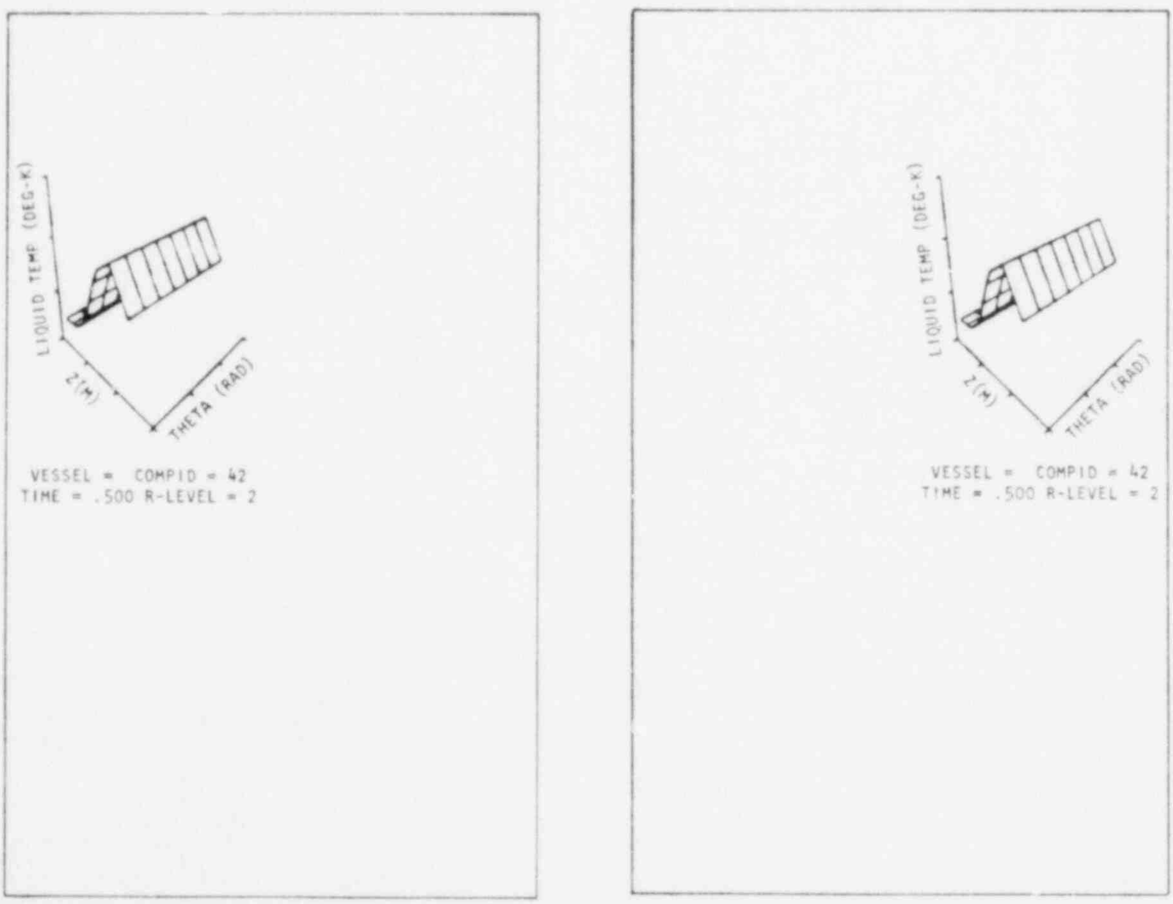
FIXED INDEPENDENT VARIABLES ARE -

```
SUMMARY OF USER SPECIFICATIONS (cont)

    X
    DEPENDENT VARIABLES ARE -
    TL
    X-CELL LIMITS    3   3
    LIMITS FOR TIME VARIABLE ARE -
       0.0000C      1.00000      .10000

? DI
Plot Request Completed
? END
All Done
```

(i) Example of use of PORT command (See Figs. 11a, 11b):



(a) (b)

Fig. 11. Examples of PORT.

```
TRAP / 1 1.5
Enter General Title
TEST CASE
Enter Problem Specification
? COM 42
? FI TH 3
? FR RA 1 5
? FR Z 1 11
? DE TL
? PORT 0. .5 .5 1.
? DI
Plot Request Completed
? PORT .5 1. .5 1.
? DI
Plot Request Completed
? EN
All Done
```

VIII. FUTURE CAPABILITIES

TRAP is written so that incorporation of new applications is relatively easy. The current package offers basic plotting capability and in attainment of this goal, time is now prime to look toward more complex and novel features. Some ideas, among an innumerable variety, include the following:

(1) level curve and level surface function contouring;

(2) vector displays of quantities like velocity over subsections of component geometries;

(3) color-coded, or otherwise shaded, function displays over two-dimensional component geometry domains; this includes shading cells at constant theta in a reactor vessel, where the shading intensity varies with numerical value of some dependent variable (as temperature);

(4) expanded macro component plots, where entire reactor systems, or subsystems, are illustrated with or without superposition of physical variable values (or color-coded representations).

Users are encouraged to submit other ideas that would expand further TRAP applications and versatility.

ACKNOWLEDGEMENTS

Appreciation is extended to all members of Q-6 and Q-9 for their valuable comments during the development and testing of TRAP. Special note is due Jim Sicilian and Rich Pryor for their continual involvement in the progress of the program. Recognition is also due to Sandy Roybal for typing the manuscript.

REFERENCES

1. "TRAC-PIA: An Advanced Best-Estimate Computer Program for PWR LOCA Analysis," Los Alamos Scientific Laboratory report NUREG/CR-0665, LA-7777-MS, (May 1979).

2. K. A. Taggart and D. R. Liles, "POST: A Postprocessor Computer Code for Producing Three Dimensional Movies of Two-Phase Flow in a Reactor Vessel," Los Alamos Scientific Laboratory report LA-NUREG-6954-MS, (September 1977).

3. DISSPLA USERS MANUAL, ISSCO, San Diego, California, Sixth Printing (1978).

4. M. Turner, "EXCON: Graphics File Merge/Convert Program," (to be published).

5. R. Kellner and A. Walker, "SCAN", LTSS Utility Routine, Los Alamos Scientific Laboratory, LTSS-515, CCF Library, (July 1978).

6. R. Kellner, "CGSGIVE", LTSS Graphics Utility, Los Alamos Scientific Laboratory, J725, CCF Library, (January 1979).

7. J. Ferguson, "Multivariable Curve Interpolation," J. ACM (April 1964).

DISTRIBUTION

	Copies
Nuclear Regulatory Commission, R4, Bethesda, Maryland	388
Technical Information Center, Oak Ridge, Tennessee	2
Los Alamos Scientific Laboratory, Los Alamos, New Mexico	50
	440

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980—677-115/157

Available from

GPO Sales Program
Division of Technical Information and Document Control
US Nuclear Regulatory Commission
Washington, DC 20555

and

National Technical Information Service
Springfield, VA 22161

)OCUMENT CONTROL DESK

016

;5