NIST United States Department of Commerce
National Institute of Standards and Technology

*NIST Special Publication 798*

# *A Users Guide for RAPID, Version 2.3*

*Richard D. Peacock, J. Newton Breese, and C. Lynn Forney*

NIST Special Publication 798

# A Users Guide for RAPID, Version 2.3

Richard D. Peacock
J. Newton Breese
C. Lynn Forney

Center for Fire Research
National Engineering Laboratory
National Institute of Standards
and Technology
Gaithersburg, MD 20899

Supersedes NBS SP 722

January 1991

**U.S. Department of Commerce**
Robert A. Mosbacher, Secretary

National Institute of Standards and Technology
John W. Lyons, Director

National Institute of Standards
and Technology
Special Publication 798
Natl. Inst. Stand. Technol.
Spec. Publ. 798
112 pages (Jan. 1991)
CODEN: NSPUE2

U.S. Government Printing Office
Washington: 1991

For sale by the Superintendent
of Documents
U.S. Government Printing Office
Washington, DC 20402

# CONTENTS

## Nomenclature

$A$ area, $m^2$

$a,b,c,k$ emperically determined coefficient

$D$ diameter (m)

$\dot{E}$ energy transfer rate, kW

$g$ gravitational constant, 9.8 $m/s^2$

$H$ heat of combustion of fuel in terms of a volume of oxygen consumed at STP, $kJ/m^3$ (default value is 17010 $kJ/m^3$) or height, m

$M$ calculated meter reading

$\dot{m}$ mass flow rate or mass loss rate, kg/s

$OD$ optical density, $m^{-1}$

$P$ fraction of full transmission

$\dot{q}$ rate of heat release, kW

$R$ is any recorded output,

$r_0$ stoichiometric oxygen/fuel mass ratio

$S$ sensitivity in Pa per output unit

$T$ temperature, K

$t$ time, s

$V$ volume flow rate, $m^3/s$

$v$ gas velocity, m/s

$W$ the width of the opening, m or molecular weights of input air, input fuel, and exhaust gas

$x$ concentration expressed as a mole fraction

$\alpha$ thermal diffusivity of the surface material, $m^2/s$ or expansion factor for fraction of air depleted of oxygen. For x(O2) greater than zero, $\alpha=1.$ for x(O2) equal zero, $\alpha$=-1./$\Delta$nu

$\beta$ stoichiometric factor for rate of heat release calculation, dimensionless (default = 1.5)

$\Delta_{hc}$ net heat of combustion, kJ/kg

$\Delta P_z$ pressure difference at height z with respect to ambient static pressure, Pa

$\varepsilon$ emissivity

$\kappa$ thermal conductivity of surface, kW/m°C

$\nu$ stoichiometric coefficients of water, carbon dioxide, carbon monoxide, oxygen, and fuel in the reaction equation

$\phi$ oxygen depletion factor, corrected for CO and $CO_2$

$\rho$ density, $kg/m^3$

$\sigma$ Stefan-Boltzmann constant ($5.667 \times 10^{-11}$ $kW/(m^2K^4)$)

# A Users Guide for RAPID, Version 2.3

## Reduction Algorithms for the Presentation of Incremental Fire Data

Richard D. Peacock, J. Newton Breese, and C. Lynn Forney

## Abstract

The voluminous amount of data that can be collected by automatic data acquisition systems during large scale fire tests requires the use of a digital computer for the reduction of data. RAPID is a stand-alone program specifically designed to convert raw instrument voltages collected during such tests into a meaningful form. The reduced data can also be used alone or in combinations to obtain derived quantities. The program is written with the ability to accept data from a user defined data acquisition system and to check the correctness of data being analyzed. The data can be converted into meaningful scientific units and then presented in tabular or printer plot form, or stored for further processing. This guide provides detailed instructions for the use of the program.

## 1 INTRODUCTION

In 1968, the Building Research Division of the National Bureau of Standards (NBS) approached the Computer Services Division of NBS with a proposal concerning the design of a series of computer programs to facilitate the analysis of automatically recorded data. During the following 2 years, a system of programs called SPEED (Systematic Plotting and Evaluation of Enumerated Data) was developed and tested [1][1]. This system was announced at the Ninth Annual Technical Symposium of the Association for Computing Machinery and in an article in Computer Graphics. The following paragraphs, quoted from the Computer Graphics article, indicate the need for SPEED is still valid:

> “The use of digital scanning systems offers several advantages to the research scientist. First, their rapid recording capabilities allow for more complete data sampling. Second, automatically recorded data is more accurate than data that has been recorded manually.”
>
> “These advantages are however, to some extent, counterbalanced by several problems which arise. Two problems are caused by the large volume of recorded data. First, it is difficult, if not impossible, to process large volumes of data by hand. Thus, the scientist finds it necessary to make use of the computer. Unfortunately, he is often unfamiliar with the capabilities and limitations of this device. Second, when presented

[1] Numbers in brackets refer to literature references list at the end of this report in section 10.

> with a large volume of data, it is often difficult for the scientist to rapidly interpret the broad characteristics of general trends that may be present. Two other problems arise in the form in which the data are recorded. The data are generally recorded in millivolts rather than standard units. Thus some conversion process, usually a linear transformation, is required. Furthermore, the recorded data are not usually directly compatible with computers. In order for a computer to read this data some special computer program must be used to read this data in the recorded form and translate it into the internal computer representation."

During the years since its announcement, SPEED has been widely used at NBS (now the National Institute of Standards and Technology) and other computer installations and has been rewritten once to provide new features and a standardized system of programs with current documentation [2]. RAPID (**R**eduction **A**lgorithms for the **P**resentation of **I**ncremental Fire **D**ata) is a stand-alone program that employs the software developed for the PLOT2 phase of SPEED2. In addition, it has been expanded and specifically designed to convert raw fire test data into meaningful units. The reduced data can be used alone or in combinations to obtain derived quantities.

This report gives detailed instructions for the use of the program and describes the implementation of the various calculations available. Section 2 provides instructions on the use of the program. Sections 3 and 4 detail the input data which must be provided to run the program. Section 5 presents an example of using the program to analyze a simple set of test data. In the remaining sections, the various calculations available to the user are described.

Extensive input data are required to analyze a typical large-scale room fire test. To structure the discussion of these inputs (and make tracing of the inevitable errors resulting from incorrect input data), the input is divided into nine parts labelled A through I. The types of data in each of these nine parts is summarized below:

Parts A and B: Data recorded by the data acquisition system prepared in the format recorded by the data acquisition system.

Part C: Input and output control.

Part D: Linear data transformations.

Part E: Description of the data acquisition system.

Parts F and G: Summary of data recorded by the data acquisition system (allows identification and correction of instrument calibrations and faulty instrument data).

Part H: Nontrivial transformations of the data matrix (this part forms the bulk of the fire-specific calculations in RAPID). This section is so lengthly that it has been separated into three sub-parts, utility, basic, and complex commands.

Part I: Graphical output control.

# 2 Using RAPID

This section provides details on the use of RAPID on the CFR Concurrent 3252 Minicomputer and on IBM PS/2 personal computers. The process will be similar on other computers, with different operating systems commands to load and execute the programs.

## 2.1 The RAPID Configuration File

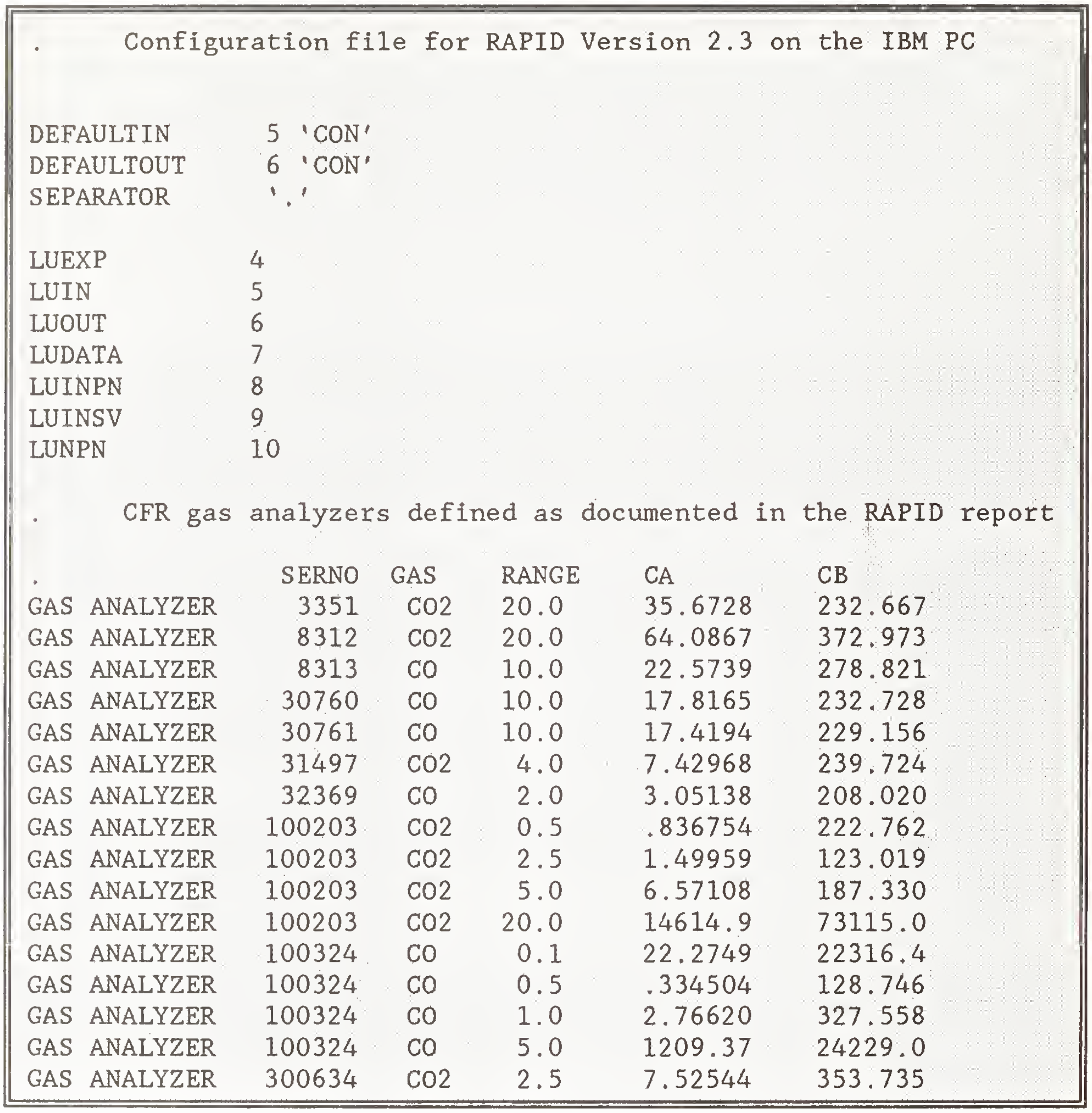
RAPID obtains information particular to each installation site and computer from a special configuration file. This file, named RCONFIG, is read each time the program is executed to initialize appropriate files for input and output and to describe the gas analysis equipment available at a particular site. The format of this file is detailed below along with its implementation on an IBM-PC (PC) at the Center for Fire Research. This may have to be modified for other installations.

The first two entries in the configuration file specify the default input and output device logical units and/or file names for the computer running RAPID. For the PC implementation, these devices are the computer keyboard (device STDIN) and screen (device STDOUT) on logical units 5 and 6.

The third entry specifies the character to be used to separate the base of a test file name from the three character extension used to identify each type of file (for instance, DTA for raw input data files). On the PC, this separator is a period.

The second group of entries specifies the logical units used by the program by default to assign to the various input and output files as detailed in the section on default file naming conventions, section 2.4.

For each gas analyzer to be predefined for a given installation of RAPID, a line is included in the configuration file which specifies the serial number of the analyzer, the type of gas analyzed, the range of the analyzer, and the calibration of the analyzer. For additional details, see the section on GAS% in section 8.1



```
.     Configuration file for RAPID Version 2.3 on the IBM PC


DEFAULTIN       5 'CON'
DEFAULTOUT      6 'CON'
SEPARATOR       '.'

LUEXP          4
LUIN           5
LUOUT          6
LUDATA         7
LUINPN         8
LUINSV         9
LUNPN          10

.     CFR gas analyzers defined as documented in the RAPID report

.               SERNO  GAS    RANGE    CA         CB
GAS ANALYZER     3351   CO2   20.0     35.6728    232.667
GAS ANALYZER     8312   CO2   20.0     64.0867    372.973
GAS ANALYZER     8313   CO    10.0     22.5739    278.821
GAS ANALYZER    30760   CO    10.0     17.8165    232.728
GAS ANALYZER    30761   CO    10.0     17.4194    229.156
GAS ANALYZER    31497   CO2    4.0     7.42968    239.724
GAS ANALYZER    32369   CO     2.0     3.05138    208.020
GAS ANALYZER   100203   CO2    0.5     .836754    222.762
GAS ANALYZER   100203   CO2    2.5     1.49959    123.019
GAS ANALYZER   100203   CO2    5.0     6.57108    187.330
GAS ANALYZER   100203   CO2   20.0     14614.9    73115.0
GAS ANALYZER   100324   CO     0.1     22.2749    22316.4
GAS ANALYZER   100324   CO     0.5     .334504    128.746
GAS ANALYZER   100324   CO     1.0     2.76620    327.558
GAS ANALYZER   100324   CO     5.0     1209.37    24229.0
GAS ANALYZER   300634   CO2    2.5     7.52544    353.735
```

## 2.2 Executing the Program

### 2.2.1 PC Version

The command to execute the program on the PS/2 is very similar: at the prompt enter -

```
RAPID <RAPID_command_line_option>
```

Note that the "version" specification is not available on the PC.

Printing output is slightly more complicated on the PC, the FORTRAN carriage control characters (found at the beginning of each line of output) should be converted to the appropriate single-, double-, and triple-line feeds, page feeds, etc., used by the specific printer for which the document is intended.

### 2.2.2 Concurrent 3252

On the Center for Fire Research Concurrent 3252, RAPID exists in four versions (for four different maximum data set sizes). The default program (RAPIDM) has a data set size of 400 columns (instruments or channels) by 400 rows (scans or readings). RAPIDS is slightly smaller (200 columns by 300 rows) and is the version used on the PS-2. RAPIDL has a larger data set size of 400 by 1000, and the fourth version, RAPIDXL has a data set size of 400 by 2000. The program limits the maximum number of columns to 1000 and places no limit on the number of rows that can be used. The memory available on individual computers will influence the size of the matrix used.

To run any version of RAPID, load one of the versions, assign logical unit 1 as shown and start the program (* is the Concurrent prompt character):

```
*LOAD RAPIDM                              (or RAPIDS, or RAPIDL, or RAPIDXL)
*AS 1,<RAPID_command_line_options>
*START
```

where <RAPID_command_line_options> is a one record file (record length up to 66 characters) that contains one of three possible options (described in sec. 2.3 below).

To make execution of the program on the Concurrent 3252 easier, a command substitution system (CSS) file has been created. When using the program there, at the prompt (*) simply enter -

```
RAPID <RAPID_command_line_option>,<version_of_RAPID_to_use>
```

where <RAPID_command_line_option> is one of the three options (described in sec. 2.3 below) and <version_of_RAPID_to_use> is one of {S,M,L,XL}. Note that if <version_of_RAPID_to_use> is left blank, M is the default.

Normally the output from the execution is left in the disk file <fn>.SAV. On the 3252, a hard copy of the output (from the printer) can be produced with the statement

```
PRINT <fn>.SAV,VFC
```

## 2.3 Command Line Options

There are three command line options:

1) <fn>
2) CREATE FRD <fn>
3) EXPLICIT <fn>

Options 1 and 2 depend strictly on a default file-naming scheme. To use this scheme, <fn> represents a test i.d. (actually the filename used for the test without the extension). The program assumes that all the files needed, or expected to be produced, will have that filename along with several default extensions (described below).

For option 3, <fn> may or may not include an extension. If the extension is included, it is used. If the extension is not included, a default extension is assumed.

### 2.4 Default Extension Naming Scheme

By using a specific set of "standard" filename extensions, you can take advantage of some easier ways of getting your data reduction done. The list of extensions used as defaults by RAPID (with their definitions) are as follows:

.CTL Data file containing the input control commands for the execution of RAPID.

.SAV File of the printed output from the execution of RAPID.

.DTA Input data file containing the raw unformatted data as recorded by the data acquisition system. It is assumed that this file exists as a normal text file for the machine running RAPID.

.FRD Output data file containing formatted raw data prepared from the data acquisition system file (the .DTA file). It is created by specifying INSAVE nonzero on RAPID data input C1. It may also be used as an input data file in subsequent runs by setting INTYPE = 2 on data input C1.

.INS If INSAVE is nonzero (RAPID data input C1) and INTYPE = 2, <fn>.FRD is assumed to be the input data file and <fn>.INS becomes the alternate default name of the created formatted data file (see .FRD, above).

.RED Output data file of the transformed data written after all calculations have been preformed on the data. It is created by specifying NPNCH nonzero on RAPID data input C2. It may also be used as an input data file in subsequent runs by setting INTYPE = 1 on data input C1.

.NPN If NPNCH is nonzero (RAPID data input C2) and INTYPE = 1 (RAPID data input C1), <fn>.RED is assumed to be the input data file and <fn>.NPN becomes the alternate default name of the created transformed data file (see .RED, above).

.IPN

If INPNCH is nonzero (RAPID data input C1), <fn>.IPN is created to contain a copy of the input data. <fn>.IPN was originally used to store the input data as it was read from magnetic tape.

.FSC

When the "CREATE FRD" option is specified on the command line, <fn'>.FSC is assumed to be the command file used to control the creation of the formatted raw data file, <fn>.FRD. <fn'> is constructed by stripping any trailing digits from <fn>. Thus, if a series of tests is performed such that the filenames used are in the form -

```
<fn> :: <prefix><suffix>
<prefix> :: <prefix><chars><char> | <char>
<chars> :: <char> | <digit> | O/
<char> :: a | b | ... | z
<suffix> :: <digit><suffix> | <digit>
<digit> :: 0 | 1 | 2 | ... | 9 | O/
```

and 1 ≤ LEN(<fn>) ≤ 8, then <fn'> is the longest <prefix> that can be found in <fn>. Consequently, <fn'>.FSC can then be used for all the tests in the series (as long as the instrument list remains the same), to generate <fn>.FRD.

.EXP

When the "EXPLICIT" option is specified on the command line, if <fn> does not include an extension, ".EXP" is the assumed extension. <fn>.EXP (or <fn>, if the extension is included) is the name of a file that will be read to find out what other logical units will be explicitly assigned to which explicitly named files. The rules for explicitly assigning files are discussed later.

### 2.5 Other Niceties to Make Life With RAPID Easier

One additional program, CCLIST, is also available. It is used to annotate the input control file images with created instrument numbers and locations. Its use has proven invaluable when creating (and especially modifying) the input control file. To use, at the prompt simply enter

```
        CCLIST <fn.ext>,<output_file_or_device>
```

where <fn.ext> is the name of the control command file and <output_file_or_device> is the name of an output file or device such as the printer or the console. The command to execute CCLIST is the same for both the Concurrent and the PS/2.

CCLIST produces a two column listing of the RAPID control commands. The left column will show the places where the program will create channels (by listing the relative order of creation preceded by a dollar sign [$]) and arrows (-->) indicating where created channel numbers are used as input. The right column is a listing of the control command file as it exists.

## 2.6 Gas Analysis Conversion Constants

A number of gas analyzers are predefined in the program and are available for use by the command "GAS%" as the curve fit identification number as detailed in section 8.1. A listing of the predefined analyzers and their constants are shown below:

| ID | Analyzer serial number | Type of gas analyzed | Full-scale range | Analyzer calibration constants | |
|---|---|---|---|---|---|
| 1 | 3351 | CO2 | 20.0 | 35.6728 | 232.667 |
| 2 | 8312 | CO2 | 20.0 | 64.0867 | 372.973 |
| 3 | 8313 | CO | 10.0 | 22.5739 | 278.821 |
| 4 | 30760 | CO | 10.0 | 17.8165 | 232.728 |
| 5 | 30761 | CO | 10.0 | 17.4194 | 229.156 |
| 6 | 31497 | CO2 | 4.0 | 7.42968 | 239.724 |
| 7 | 32369 | CO | 2.0 | 3.05138 | 208.020 |
| 8 | 100203 | CO2 | 0.5 | 0.836754 | 222.762 |
| 9 | 100203 | CO2 | 2.5 | 1.49959 | 123.019 |
| 10 | 100203 | CO2 | 5.0 | 6.57108 | 187.330 |
| 11 | 100203 | CO2 | 20.0 | 14614.9 | 73115.0 |
| 12 | 100324 | CO | 0.1 | 22.2749 | 22316.4 |
| 13 | 100324 | CO | 0.5 | 0.334504 | 128.746 |
| 14 | 100324 | CO | 1.0 | 2.76620 | 327.558 |
| 15 | 100324 | CO | 5.0 | 1209.37 | 24229.0 |
| 16 | 300634 | CO2 | 2.5 | 7.52544 | 353.735 |
| 17 | 300634 | CO2 | 15.0 | 12.2465 | 141.164 |
| 18 | 300635 | CO | 1.0 | 6.34414 | 685.198 |
| 19 | 300635 | CO | 7.0 | 5.00492 | 132.479 |
| 20 | 30759 | CO | 10.0 | 17.1453 | 226.150 |
| 21 | 32062 | CO | 2.0 | 2.71598 | 191.773 |

| | | | | | |
|---|---|---|---|---|---|
| 22 | 32371 | CO2 | 20.0 | 39.0345 | 249.152 |
| 23 | 34537 | CO | 15.0 | 20.5896 | 193.337 |
| 24 | 34539 | CO2 | 20.0 | 29.3901 | 202.437 |
| 25 | 34677 | CO2 | 10.0 | 16.7002 | 221.373 |
| 26 | 34865 | CO2 | 10.0 | 13.7266 | 193.402 |
| 27 | 34540 | CO2 | 20.0 | 32.8684 | 219.574 |
| 28 | 34391 | CO | 10.0 | 17.6194 | 230.505 |
| 29 | 34538 | CO | 15.0 | 20.0793 | 189.892 |
| 30 | 32372 | CO2 | 20.0 | 29.7035 | 203.892 |
| 31 | 34678 | CO2 | 10.0 | 16.6299 | 220.467 |
| 32 | 34753 | CO | 5.0 | 6.4381 | 185.106 |
| 33 | 103522 | CO2 | 2.5 | 4.4600 | 231.000 |
| 34 | 31445 | CO2 | 20.0 | 37.3761 | 237.640 |
| 35 | 32098 | CO | 1.0 | 1.2718 | 182.966 |
| 36 | 100203 | CO2 | 5.0 | 6.3243 | 181.815 |
| 37 | 100324 | CO | 1.0 | 2.1355 | 261.877 |
| 38 | 101403 | CO2 | 5.0 | 6.1889 | 8.972 |
| 39 | 101403 | CO2 | 20.0 | 12.3116 | 6.180 |
| 40 | 101404 | CO | 1.0 | 2.4985 | 15.073 |
| 41 | 101404 | CO | 5.0 | 2.5896 | 5.774 |
| 42 | 32796 | CO | 1.0 | 1.7470 | 229.930 |

# 3 The Test Data File

A minimum of two data files are required for analysis of test data using RAPID: the test data (collected by an automatic data acquisition system), and a control data file to supervise the execution of the program. This section describes different formats for the file of data collected by a data acquisition system. Section 4 describes the control data file.

Typically, a data acquisition system will store the collected test results on some magnetic media such as tape or disk. Since the output of each system is different, details of transferring the data from the data acquisition system to the computer running RAPID is beyond the scope of this report. It is assumed that the file of data collected by the data acquisition system is a normal text file for the computer running RAPID. For some systems, this may involve some translation from an internal format of the data acquisition system to a text file. Several examples of these formats are illustrated below.

The test data file and control data files consist of several distinct sections of data. For ease of discussion, each of these is assigned a letter designation of Part A through Part I. In addition, each line of input is assigned a unique number as a suffix to the Part designation. Thus, the first input for Part A data is input A1.

## 3.1 Input Data in Data-Acquisition-System-Dependent Formats

For Part A, the test data file consists of the data recorded by the data acquisition system prepared in the format recorded by a data acquisition system. Different formats, such as those described in table 1 or as defined in the control data file, are possible. If the data images were prepared by an earlier run of RAPID (by setting INPNCH greater than zero), the set of data images produced should be in the proper format.

Table 1. Pre-defined data formats for input to RAPID

INTYPE = 1: reduced data format input data file
INTYPE = 2: pre-processed raw data input data file in reduced data format
INTYPE = 3: VIDAR 5400 series input data file format
INTYPE = 4: VIDAR 5400 series magnetic tape format
INTYPE = 5: Hewlett-Packard 3497 input data file format
INTYPE = 7: VIDAR Autodata 10 series input data file format
INTYPE = 9: VIDAR Autodata 9 series input data file format
INTYPE = 10: VIDAR Autodata 9 series magnetic tape format

| INTYPE | Reading | Time | End of record | | End of file | |
|---|---|---|---|---|---|---|
| | | | Tape | Data File | Tape | Data File |
| 1 | s.vvvvvEsee | s.vvvvvEsee | | none | | 77*ƀ999 |
| 2 | s.vvvvvEsee | s.vvvvvEsee | | none | | 77*ƀ999 |
| 3 | cccsvvvvveƀƀ | sss6ssssssƀƀ | | ƀX | | FILEND |
| 4 | cccsvvvvveƀƀ | sss6ssssssƀƀ | EOR | | EOF | |
| 5 | CcccƀsvvvvvvvvEeeeƀƀX | aaaaƀddaaaaaaaaaƀhh:mm:ss | | EOR | | EOF |
| 7 | CcccƀsvvvvvvvvEeeeƀƀX | aaaaƀddaaaaaaaaaƀhh:mm:ss | | EOR | | EOF |
| 9 | cccƀsvvvvvvaaaeX | dddhh:mm:ssƀƀƀƀX | | ƀX | | FILEND |
| 10 | cccƀsvvvvvvaaaeX | dd:hh:mm:ssaaaaX | EOR | | EOF | |

| | | |
|---|---|---|
| a - any character | n - a numeric digit | C - the character "C" |
| o - an overflow indicator | r - value of reading | E - the character "E" |
| c - a channel number digit | d - days portion of a time reading | X - the character "X" |
| s - sign of a reading | h - hours portion of a time reading | EOR - magnetic tape end of record |
| ƀ - blank | m - minutes portion of a time reading | EOF - magnetic tape end of file |
| v - magnitude of reading | s - seconds portion of a time reading | |
| e - exponent of reading | | |

*Example*

```
TIME 25 Aug 1989 13:17:53 X    Data  SCAN
C011  2.75936E-03  XC012  9.98200E-04  XC013 -1.00000E-06  XC014  1.08044E-02 X
C015  7.08000E-05  XC016  6.74730E-02  XC017  8.34729E-01  XC018  1.99436E+00 X
C019  5.47560E-03  XC020  8.93200E-04  XC021  8.87500E-04  XC022  8.90000E-04 X
EOR
EOF
```

## 3.2 Input Data in Data-Acquisition-System-Independent Format

Alternatively, the test data file may consist of the data recorded by the data acquisition system in a format that is independent of the recording system used to collect the data. This data is typically generated by an earlier run of RAPID when the control parameters INSAVE and/or NPNCH are greater than zero.

Part B data inputs are read only if parameter INTYPE (Data Input C1) is equal to 2 or 4.

**Input:** B1

**Variables:** *NPTS, KH, NAME1, *, NAME2*
**Format:** 2I6, A6, A1, A60

*NPTS* is the number of data points for this instrument.
*KH* is the instrument number.
*NAME1* is a six character abbreviated instrument identification.
* is the ASCII character '*'.
*NAME2* is a 60 character description of the instrument.

Note that the string '999' in columns 78-80 terminates the reading of Part B data inputs.

**Input:** B2

**Variables:** *REED(1),REED(2) ,..., REED(i), 1 <= i <= NPTS*
**Format:** 7E11.5

*REED(i)* is a single data or time reading.

Note that as many B2 inputs as necessary should be entered until all NPTS data points have been entered.

*Example*

```
    21  4011CO2   *CO2 Carbon Dioxide concentration                   (Vol %)
0.38357E-020.25693E-020.18335E-020.14006E-020.10898E-020.84953E-030.67168E-03
0.53167E-030.39878E+000.34335E+000.28012E+000.22961E+000.19193E+000.16208E+00
0.13883E+000.12018E+000.10548E+000.94008E-010.84671E-010.77969E-010.73271E-01
                                                                             999
```

## 4 The RAPID Control File

The RAPID control file is simply a file which contains information read by the program which directs the execution of the program. This information includes options which specify the format and location of input data (data collected by a data acquisition system), direct the creation of output files, and provides commands to the program for any data analysis to be performed. This section provides details on the information contained in this control file. Section 2 provides a default naming scheme for the data files which allows easy, consistent, and automatic naming of all of these data files. For reference, the RAPID control file is the ".CTL" file discussed in section 2.

## 4.1 Input and Output Control

**Input:** C1

**Variables:** *INTYPE, INPRT, INPNCH, INSTOP, INERR, INSKIP, INSAVE, INTEST*
**Format:** 8I5

This input contains parameters which control the input. The various possibilities and their meanings are:

*INTYPE* defines the data collection hardware and the format of the data recorded by the data collection system. The possible values for INTYPE are as follows: 0, read Part E data inputs to specify a special data acquisition system; 1, reduced data format input images (the ".RED" file format described in section 2.4); 2, pre-processed raw data input images in reduced data format (the ".FRD" file format described in sections 2.4 and 3.2); 3, VIDAR 5400 series input image format; 4, VIDAR 5400 series magnetic tape format [not available in this version]; 5, Hewlett-Packard 9836 input image format; 6, Hewlett-Packard 9836 magnetic tape format [not available in this version]; 7, VIDAR Autodata 10 series input image format; 8, VIDAR Autodata 10 series magnetic tape format [not available in this version]; 9, VIDAR Autodata 9 series input image format; 10, VIDAR Autodata 9 series magnetic tape format [not available in this version].

*INPRT* directs printing of data as recorded by a data acquisition system. If INPRT is equal to zero, no data is printed. If INPRT is greater than zero, INPRT specifies the maximum number of data records to be printed. If INPRT is equal to -1, all data records are printed. If INPRT is equal to -2, only data records that contain errors are printed.

*INPNCH* directs the output (echo) of the data used as input to secondary storage (disk file, magnetic tape, printer, etc.). If INPNCH is nonzero, all data records are written to the device or file specified by the user, or if not specified, to a default device or file. (See sec. 2, Using RAPID).

*INSTOP* determines if the program will continue execution once the input data has been processed. If INSTOP is non-zero, RAPID will stop execution after processing the input data recorded by the data acquisition system.

*INERR* specifies the maximum number of error messages to be printed during processing of data recorded by a data acquisition system.

*INSKIP* determines whether input records from the data collection system will be selectively skipped during processing of the information from the data collection system. This feature may be used to ignore header information or erroneous data outputs. If INSKIP is non-zero, read Part F data inputs to specify records of input data to be skipped.

*INSAVE* directs the output of interpreted raw data to secondary storage (disk file, magnetic tape, printer, etc.). If INSAVE is nonzero, all data records not skipped (see INSKIP are written to the device or file specified by the user, or if not specified, to a default device or file. (See sec. 2, Using RAPID.) This "formatted raw data" may be identified and used as input to subsequent runs by setting INTYPE equal to 2. The format of the raw data saved is the same as any reduced data saved by setting NPNCH greater than zero (see input C2 below). See Part B and NPDI[2] [4].

*INTEST* determines the fate of test number identification included with the input data. See NPDI [5]. If 0, include the test number (if it exists) on input, if less than 0, delete the test number (if it exists) on input.

**Input:** C2

**Variables:** *NTEST, NPRT, NPNCH, NPLOT, NCORR, NERR*
**Format:** 6I5

This input contains certain parameters which control actions concerning the transformed data matrix. The possibilities are:

*NTEST* allows the user to specify a test identification number for the output data file. See NPDI [5]. Numbers for NTEST between 1 and 999, inclusive specify a test number to be prefixed to reduced data instrument numbers output by setting NPNCH greater than zero. If NTEST is equal to 0, on saved output (see NPNCH) only include the test number if it already exists (from input).

*NPRT* directs printing of transformed output matrix - See NPDI [6]. If NPRT is greater than zero, print out the transformed data matrix and summary of minima, maxima, and averages. If less than zero, print out only the summary. If equal to zero, no printout.

*NPNCH* directs the output of the transformed data matrix to secondary storage (disk file, magnetic tape, printer, etc.). If NPNCH is nonzero, all data records are written to the device or file specified by the user, or if not specified, to a default device or file. (See sec. 2, Using RAPID.) This "reduced data" may be identified and used as input to subsequent runs by setting INTYPE (Input C1 above) to 1. See Part B.

*NPLOT* directs the plotting of the output data. If NPLOT is non-zero, Part I data inputs are read to generate printer plots of selected instruments.

*NCORR* specifies corrections to be made to the data matrix. If NCORR is non-zero, Part G data inputs are read to correct readings of the data matrix.

[2] Appendix A presents a series of notes on the preparation of the control inputs for RAPID. Throughout the text these are referred to as NPDI *n* where *n* is the number of the note of interest. A.*n* provides a description of NPDI *n*.

*NERR* specifies the maximum number of error messages to be generated by any one data reduction command.

**Inputs:** C3, C4

**Variables:** *TITLE(1:80), TITLE(81:120)*
**Format:** A80/A40

These two inputs specify the title of the experiment, printed at the top of all pages of output.

*Example*

```
    2    0    0    0    0    1    0
    7    1    8    1    0    5
FRCA LARGE SCALE TEST               FRCAFS07                          87.1103
FRCA A.2.B.  NON FR ROOM
```

### 4.2 Test and Instrument Descriptions

Control inputs in this section specify the instruments included in the input data set and any linear conversion to be performed on sets of instruments. Each instrument is identified with a unique number (usually assigned by the data collection equipment), an instrument type (to identify the time channel and any linear conversion to be performed on the instrument), and a description of the instrument.

**Input:** D1

**Variables:** *KH(i), ITYPE(i), NAME(i), KHPRT(i)*
**Format:** I6, I2, A66, A3

For each instrument included in the data matrix, there must be a input of this form defining the instrument number, *KH(i)*; the instrument type, *ITYPE(i)*; and the instrument name, *NAME(i)*. The instrument number, KH(i), is either the channel number assigned by the data acquisition system, or, for user created instruments, a unique number assigned by the user. NAME(i) is broken into two parts: a 6-character abbreviated ID that is printed when listing or plotting data and a 60-character description. If *KHPRT(i)* is non-blank, the transformed instrument values will NOT be printed. Note that all the channels may be skipped by setting NPRT <= 0 on Data Input C2. The number of inputs is variable, with the end signalled by a input D2, below.

**Input:** D2

**Variable:** *IEND*
**Format:** 77X, A3

If ***IEND*** is equal to 999, this input signals the end of the set of instrument defining inputs, D1 above.

**Input:** D3

**Variables:** *C(i), ADD(i), POWER(i)*
**Format:** 1X, 3F15.6

Each ***C(i)***, ***ADD(i)***, and ***POWER(i)*** represent the conversion coefficients for all instruments of type i. There are as many D3 inputs as there are different types of instruments as defined in the set of instrument inputs, D1 above. (See NPDI [7].)

**Input:** D4

**Variable:** *IEND*
**Format:** 77X, A3

If ***IEND*** is equal to 999, this input signals the end of the set of conversion coefficient inputs, D3 above.

*Example*

```
00 1Time  Elapsed time                                         (s)
11 2CO2   CO2 Carbon Dioxide concentration                 (Vol %)
   0      Beckman CO2 analyzer, linear calibration
12 2CO2   Carbon Dioxide concentration, Beckman CO2 analyzer  (Vol %)
   0      serial # 34678, 10.0 % full scale, span gas 9.22%
13 3RADMTRRadiometer serial #624416, 30" away              (kW/M²)
14 2CO    Carbon Monoxide concentration, Beckman CO analyzer  (Vol %)
   0      serial # 34753, 5.0% full scale, span gas 4.66%
                                                                    999
 1.0
 1.0
 2635.5
                                                                    999
```

### 4.3 Input Formats for Data Acquisition Systems Not Pre-Defined in RAPID

If the data acquisition system is not one of the pre-defined types (see table 1, page 11), the user may have to define the formats the program will need for the input media, the time and data readings, and the end-of-file marker. At this time, there is only one format for specifying the input media. The syntax for the format is below:

```
INPUT=DATA IMAGES, CHANNELS PER LINE=<n>
```

where n is any integer number greater than zero, and the number is the maximum number of channels found on a single input image. As is noted, the format is a specification for data images. Data added from mass storage or tape files, and data transferred from remote terminals are considered DATA IMAGES. The syntax for defining the data readings, time readings, and end-of-record and end-of-file markers is the same for all four:

```
READING=<pattern>     or
TIME=<pattern>        or
EOR=<pattern>         or
EOR=EOR               or
EOF=<pattern>         or
EOF=EOF
```

where pattern is defined by

$$[N_1]^*(C_1)[N_2]^*(C_2)[N_3]^*(C_3) \ldots [N_{m-2}]^*(C_{m-2})[N_{m-1}]^*(C_{m-1})[N_m]^*(C_m)$$

and

$N_1, N_2, N_3 \ldots N_{m-2}, N_{m-1}, N_m$ = any integer greater than or equal to 1.

$C_1, C_2, C_3 \ldots C_{m-2}, C_{m-1}, C_m$ = one or more of the following character specifications:

| | |
|---|---|
| S | the seconds portion of the time reading |
| M | the minutes portion of the time reading |
| H | the hours portion of the time reading |
| D | the days portion of the time reading |
| N | any numeric digit (0-9) |
| A | any alphanumeric character |
| C | a channel number digit |
| +<char> | the character <char> used to identify a positive reading |
| -<char> | the character <char> used to identify a negative reading |
| V | a numeric digit of the value of the instrument reading |
| R | a numeric digit of the value of the instrument reading, possibly with an embedded decimal point |
| E | a numeric digit of the exponent of the instrument reading |
| O<char> | the character <char> used to identify an overflow in the instrument reading |
| K<char> | a special single character <char> |

If several possibilities exist for a single character, then all possibilities are placed within the parentheses. For instance, if a single character is used to indicate +, -, or overflow, it might be coded as (+1-2O9) defining the plus indicator as 1, the minus indicator as 2, and the overflow indicator as 9.

Consider a reading as follows: three digits of channel, a single indicating the sign of the reading or overflow, five characters indicating the value of the reading, a single character exponent and two spaces (for example, "0011324654  ". It could be coded as:

READING=(C)(C)(C)(+1-2O9)(V)(V)(V)(V)(V)(E)(K )(K )

or equally

READING=3*(C)(+1-2O9)5*(V)(E)2*(K ).

The forms EOR=EOR and EOF=EOF are used for magnetic tape media and indicate, respectively, that the data records are separated by physical record gaps on the tape and that there is a physical end-of-file mark on the tape.

**Inputs:** E1-E5

**Variables:** *INE1, INE2, INE3, INE4, INE5*
**Format:** A80/A80/A80/A80/A80

These five lines of input specify the format of data collected by the data acquisition system. The five inputs are as follows: *INE1* specifies the input media definition described above. *INE2* specifies the reading definition described above. *INE3* specifies the time definition described above. *INE4* specifies the end-of-record definition described above. *INE5* specifies the en-of-file definition described above.

Note that data Inputs E1 through E5 are only entered if parameter INTYPE (Data Input C1) is zero.

*Example*

```
INPUT=DATA IMAGES, CHANNELS PER LINE=4
READING=(KC)3*(C)(K )(+ --)7*(R)(KE)3*(E)2*(K )(KX)
TIME=(KT)(KI)(KM)(KE)(K )(K D)(D)9*(A)(K )2*(H)(K:)2*(M)(K:)2*(S)(K )(KX)53*(A)
EOR=(K KE)(KEKO)(KOKR)(KRK )
EOF=(KE)(KO)(KF)
```

### 4.4 Skipping Data Records on Input

Part F specifies the records to be skipped during the processing of data recorded by a data acquisition system. The records identified to be skipped are ignored on input; no translation of the skipped records is done[3].

There are two methods of describing which records are to be skipped:

1. Up to 16 different individual records can be enumerated by entering the scan number of the record to be skipped.

2. A set of “skip/keep” specifications, defining the records to be skipped and kept can be input.

For method 2, the pattern format is defined as follows:

$$\text{SKIP}=(C_{1,1}N_{1,1}C_{1,2}N_{1,2}\ldots C_{1,n}N_{1,n})R_1\ (C_{2,1}N_{2,1}C_{2,2}N_{2,2}\ldots C_{2,n}N_{2,n})R_2 \\ \ldots\ (C_{m,1}N_{m,1}C_{m,2}N_{m,2}\ldots C_{m,n}N_{m,n})R_m$$

where a “skip/keep” specification is a list of sub-specifications within a pair of parentheses followed by an end-of-specification record number, and

$C_{i,j}$ = S, Z, R, K or F
where, S stands for Skip
Z stands for skip (after saving Zeros)
R stands for skip (after saving Ranges or spans)
K stands for Keep
F stands for Final record

$N_{i,j}$ = any number >= 1
where the number is an integer indicating the number of times $C_{i,j}$ is to be repeated

$R_i$ = any number >= 1
where the number is an integer indicating the last record number on which the sub-specifications are impacted.

As can be seen, any number of sub-specifications can be used within a pair of parentheses. For example, if the user wishes to skip two records and keep one record from record 1 to record 200, it would be coded as

SKIP=(S2K1)200.

[3] The exception to this is when the special forms of "skip" are used, i.e., when zeros and ranges (spans) are requested. In those cases, the appropriate values are stored in arrays other than the data matrix. Therefore, the data matrix still appears as if those records were skipped.

Consider the coding for the following requirements — The user wishes to keep records 1 to 5; skip 2 records and keep 1 record for records 6 to 150; then keep every record for records 151 to 599; record 600 is to be the last record processed:

SKIP=(K1)5 (S2K1)150 (K1)599 (F1)600

Two special forms of the skip also exist. They are "zero" and "range" (or "span"). If "Z" or "R" is used instead of "S", a value is stored in a special array for each of the instruments corresponding to the records specified before the record is skipped. If more than one record is specified as a "zero" record, the value stored is the minimum value from the specified records. Likewise, the value stored for "range" records is the maximum value from all specified records.

Note that care should be taken to ensure that no overlaps or conflicts exist in the pattern. If conflicts exist, the first encountered specification that applies to a given input record will be used, leading to potentially unpredictable results.

Part F should only be entered if parameter INSKIP (Data Input C1) is greater than zero.

**Input:** F1a

**Variables:** ISKIP(1), ISKIP(2), . . . , ISKIP(i), 1 <= i <= 16
**Format:** 16I5

This is method 1. *ISKIP* is the number of the record to be skipped.

**Input:** F1b

**Variable:** IN
**Format:** A80

This is method 2. *IN* is the skip/keep pattern as described above.

*Example*

```
SKIP=(Z2R7)9
```

### 4.5 Corrections to the Data Matrix

Part G data inputs are read only if parameter NCORR (Data Input C2) is non-zero. As many additional sets of G1 and G2 inputs as are required may be included at this point to make the necessary corrections.

**Input:** G1

**Variables:** *IRL, IRH, ICL, ICH*
**Format:** 4I5

The variables define a low row index (IRL), a high row index (IRH), a low column index (ICL), and a high column index (ICH) to define the portion of the data matrix to be corrected. Note that there are restrictions on the values of these parameters. See discussion below.

Note that setting IRL less than zero terminates the reading of Part G data inputs.

**Input:** G2

**Variables:** *REED(i,j), i=IRL,IRH; or j=ICL,ICH*
**Format:** 8F10.0

These are the corrections to the matrix.

*Example*

```
    2    2    5    10
23.0      24.2      134.5     321.2     423.4     323.6
```

Any number of corrections may be made; however, any single correction may apply only to one single entry, consecutive entries in a single column, or consecutive entries in a single row. There are, in effect, only three valid combinations:

**Correct a single entry:** IRL is equal to IRH and ICL is equal to ICH meaning correct entry REED(IRL,ICL).

**Correct part or all of a row:** IRL is less than IRH and ICL is equal to ICH meaning correct entries REED(IRL,ICL), REED(IRL+1,ICL), ..., REED(IRH-1,ICL) and REED(IRH,ICL).

**Correct all or part of a column:** IRL is equal to IRH and ICL is less than ICH meaning correct entries REED(IRL,ICL), REED(IRL,ICL+1),..., REED(IRL,ICH-1) and REED(IRL,ICH).

Failure to satisfy any of the above restrictions will result in an error message being printed and may result in all following data inputs being out of order. Thus, particular care must be taken in the preparation of this input.

In any of the above cases, the number of entries to be corrected is

(IRH-IRL) + (ICH-ICL) + 1.

### 4.6 Non-Trivial Transformations of the Data Matrix

In Part D, above, it is possible to identify conversion constants for each instrument that allow the user to multiply, add to, and raise to a power, the value of each instrument by those constants.

However, in many cases the use of those constants is not sufficient to transform the raw data into values of use to the test analyst. Therefore, a large set of subroutines is included at this point to allow the conversion and manipulation of not only raw data, but also the combinations of converted data needed to produce the complex variety of values required for good fire test analysis.

The description of the input for Part H is rather extensive, so, in order to preserve the continuity of this document, it is detailed beginning in section 7 after the description of the remaining general control inputs.

### 4.7 Plotting

Although not always precise in the conveyance of information, printer plots can be a useful tool to the test analyst.

Three forms of plots can be used here:

1. PLOT Nx Ny1 Ny2 ... Nym

   where Ny1, Ny2, ..., Nym are any number of instrument numbers representing the y-axis values being plotted versus instrument number Nx, the x-axis values.

2. PLOT Nx1,Ny1 Nx2,Ny2 ... Nxm,Nym

   where Ny1, Ny2, ..., Nym are any number of instrument numbers representing the y-axis values being plotted versus instrument numbers Nx1, Nx2, ..., Nxm (respectively), the x-axis values.

3. PROFILE

   where, typically, values are plotted versus position rather than versus time.

Part I data inputs are read only if parameter NPLOT (Data Input C2) is non-zero.

**Input:** I1

**Variable:** *IN*
**Format:** A80

*IN* is one of the three forms of plots described above. If form 1 or 2 is used, and there is not enough room on one input to identify all the instrument numbers required, this input may be continued by placing a semicolon (;) at the end of the input line. See NPDI [3].

**Input:** I2

**Variable:** *GTITL*
**Format:** A80

*GTITL* is the 80 character graph title printed above the graph. If Form 3 of input I1 was used, GTITL is concatenated with the string 'PROFILE OF THE FOLLOWING CHANNELS:'

**Input:** I3

**Variables:** *JCHAN(1), JCHAN(2), . . . , JCHAN(i)* X
or, $1 <= i <= 20$
*CON(1), CON(2), . . . , CON(i)* X
**Format:** EVALU8 (NPDI [2])

Read this input only if Form 3 of input I1 is used. Two of these inputs **must** be prepared. The two inputs should consist of one each of a set of channel numbers and a set of constants. Whichever set is first is plotted against the x-axis; the second against the y-axis. *JCHAN* is the instrument number of the values to be plotted in the profile (see NPDI [3]). *CON* is a constant value. The number of constants should match the number of channels. Note: if R(t,j) is the reading of the jth instrument (JCHAN(j)) at time, t, then CON(j) and R(t,j) form an ordered pair to be plotted. The order within the pair depends on which set of values is entered first. *X* is the character string ' X', the end-of-set mark for input with the routine EVALU8.

**Input:** I4

**Variables:** *JTIME, ISCAN(1), ISCAN(2), . . . , ISCAN(k)* X 1 <= k
**Format:** EVALU8 (NPDI [2])

Read this input only if Form 3 of input I1 is used. *JTIME* is the time channel instrument number (see NPDI [3]). *ISCAN* determines the data points to be plotted in the profile plot. If ISCAN is an integer, it is the scan number of the values of the JCHAN to be used. If ISCAN is a real (has a decimal point) it is the time of the values of the JCHAN to be used. If the time cannot be exactly matched, the time nearest without going over is used. *X* is the character string ' X', the end-of-set mark for input with the routine EVALU8.

**Input:** I5

**Variables:** *XL, XH, YL, YH*
**Format:** open (NPDI [1])

This input is used for all forms of the plots. *XL* is the lower limit of the X axis. *XH* is the upper limit of the X axis. *YL* is the lower limit of the Y axis. *YH* is the upper limit of the Y axis.

**Input:** I6

**Variables:** *XBUFF, YBUFF*
**Format:** 2A40

This input is used for all forms of the plots.

*XBUFF* is the 40 character x-axis title. The title will be centered along the horizontal axis by the program.
*YBUFF* is the 40 character y-axis title. The title will be centered along the vertical axis by the program.

*Example*

```
PLOT 000 $05
HEIGHT OF HOT/COLD INTERFACE IN TARGET ROOM
0., 2500., 0., 2.5
TIME - S                                HEIGHT | M
PROFILE
PROFILE OF TEMPERATURES IN BURN ROOM DOORWAY
23  24  25  26  27  28  29  30  31  32  X
0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 X
0 30 60 90 X
0., 500., 0., 2.
TEMPERATURE - C                         HEIGHT | M
```

# 5 A Simple Example

To introduce the first time user, this section provides an in-depth example of the use of RAPID for a simple set of data. The user is guided through the creation of a RAPID input control file for the example. The output of the program for the example is shown in appendix A. As described above, two input files are required: one which includes the data to be analyzed and a control file to direct the program operation. Each of these is described in detail below.

## 5.1 The Data-Acquisition-System-Dependent Input File

For this example, we will use a fairly small scale test with only a few input channels. A sample of part of the set of input data is shown below:

```
TIME 25 Oct 1989 10:37:08 X    Zero  SCAN
C011  1.13242E-02  XC012  1.40100E-04  XC013 -3.50000E-06  XC014  6.69000E-05 X
C015 -7.11000E-05  XC016  6.41569E-02  XC017  6.33000E-04  XC018 -9.16340E-03 X
C019  6.89100E-04  XC020  4.51700E-04  XC021  4.50100E-04  XC022  6.77100E-04 X
EOR
TIME 25 Oct 1989 10:43:45 X    Span  SCAN
C011  7.41766E+00  XC012  9.35277E-02  XC013 -1.20000E-06  XC014  9.57131E-02 X
C015 -7.11000E-05  XC016  1.81330E-01  XC017 -6.66000E-04  XC018  1.93213E+00 X
C019  9.33500E-04  XC020  7.20000E-04  XC021  7.18400E-04  XC022  6.60000E-04 X
EOR
TIME 25 Oct 1989 11:19:54 X    Data  SCAN
C011  9.59106E-02  XC012  7.03200E-04  XC013 -7.00000E-07  XC014  1.22000E-04 X
C015 -6.99000E-05  XC016  2.41435E-01  XC017  8.38175E-01  XC018  2.18293E+00 X
C019  1.16300E-04  XC020  7.37500E-04  XC021  7.37700E-04  XC022  6.77000E-04 X
EOR
TIME 25 Oct 1989 11:20:02 X    Data  SCAN
C011  9.62152E-02  XC012  7.01800E-04  XC013 -1.70000E-06  XC014  9.12000E-05 X
C015 -7.08000E-05  XC016  2.41445E-01  XC017  8.38516E-01  XC018  2.19924E+00 X
C019  6.50000E-06  XC020  7.39800E-04  XC021  7.38200E-04  XC022  6.78200E-04 X
EOR
```

```
    .          .          .          .          .          .          .          .
    .          .          .          .          .          .          .          .
    .          .          .          .          .          .          .          .

TIME 25 Oct 1989 11:32:32 X     Data  SCAN
C011  9.59496E-02  XC012  7.08800E-04  XC013 -4.00000E-07  XC014  7.32000E-05 X
C015 -3.73987E+00  XC016  2.40763E-01  XC017  8.37052E-01  XC018  2.18882E+00 X
C019  4.62300E-04  XC020  7.47800E-04  XC021  7.46600E-04  XC022  6.85400E-04 X
EOR
EOF
```

Several features of this file are apparent.

- Each set of readings through the instruments (called a *scan*) begins with a time reading (for instance, `TIME 25 Oct 1989 11:32:32 X`).

- The type of scan is noted (here, three types of scans are shown: Zero, Span, and Data). This identification allows the program to identify automatically the instrument calibration.

- The readings from each channel of data follow an identical format showing the channel number, the value of the reading, and a unique character sequence to identify the end of the reading (for example, `C012  7.01800E-04  X`).

- The end of each scan and the entire test are identified by unique sequences of characters (here, `EOR` for the scan and `EOF` for the test).

### 5.2 The RAPID Control Input File

For each test, a set of control input data must be generated to direct the action of RAPID. Usually, this will be generated with a text editor on the machine where the program will be run. This file can be separated into (at least) four parts: input and output control parameters, test and instrument identification, calculations, and plot generation. Each of these is described in detail below.

The entire input file for the example case is shown below.

```
    5    0    0    0    0    1    0    0
    4   -1    1    1    0    5
Furniture Calorimeter      Test 4
19 Oct 1989
    00 1Time  Elapsed time                                          (s)
    11 2CO2   CO2 Carbon Dioxide concentration                  (Vol %)
       0      Beckman CO2 analyzer, linear calibration
    12 2CO2   Carbon Dioxide concentration, Beckman CO2 analyzer  (Vol %)
       0      serial # 34678, 10.0 % full scale, span gas 9.22%
    13 3RADMTRRadiometer serial #624416, 30" away                 (kW/M^2)
    14 2CO    Carbon Monoxide concentration, Beckman CO analyzer  (Vol %)
       0      serial # 34753, 5.0% full scale, span gas 4.66%
    15 2MARKEREvent marker                                                 (V)
    16 2WTLOSSTotal weight loss                                           (kg)
    17 202    Oxygen concentration, linear calibration                 (Vol %)
       0      Taylor O2 analyzer
    18 2GASVELBi-directional velocity probe                              (m/s)
    19 2SMOKE Smoke extinction coefficient                               (m-1)
    20 2TC BOTTemperature at bottom of stack                              (°C)
    21 2TC TOPTemperature at top of stack                                 (°C)
    22 2RMTEMPAmbient temperature at back of box                          (°C)
                                                                            999
     1.0
     1.0
     2635.5
                                                                            999
SKIP=(Z2R4)6
TIME  60 second delay from data collection start to ignition
00 0 0 -60
GAS%  Convert gas analyzers from voltage to volume percent
4
11  3      0.   10.        5.    X
12  1 31   Z    R         93.5   X
14  1 32   Z    R         95.35  X
17  3      Z    0.834673  20.9   X
TC
1
20 22 X
VELOCITY
1
18  49.74   Z   3  1  21  X
DELAY  Time for gas to travel down exhaust pipe to measuring instruments
1
11  12  14  17  18  19  20  21  22  00  9.5 X
WT-LOSS Calculate total weight loss of specimen on load platform
1
16 192.6  X
SPECIFY  Re-use the weight loss channel for smoothed weight loss calculation
16 X
SMOOTH   Weight loss readings
1
16 00 7 X
MASS-FLOW-3  Calculate mass flow of gas in exhaust stack for HRR calculation
18 0.18 21 .9 X
HEAT-RATE-2  Calculate heat release rate from stack measurements
17 $02 1 11  X
SPECIFY  Re-use the HRR channel for conversion from kW to MW
$03 X
COMPUTE  Change HRR from kW to MW
1
$03 / 1000. X
RENAME
    $1  SMOKE SMOKE EXTINCTION COEFFICIENT
    $2  MFLOW MASS FLOW RATE OF GAS THRU STACK (KG/S)
    $4  HRR   HEAT RELEASE RATE THRU STACK (MW)
                                                                           999
END
PLOT 00 $04
Heat release rate measured in exhaust stack
0. 2500. 0. 1.00
Time - s                                Heat release rate | MW
```

### 5.2.1 Input and Output Control Parameters

The first four lines (detailed in sec. 4.1) of the input file describe the test and provide various input and output parameters to direct the program to accept input in particular formats and produce certain printed or stored output. The following information should be determined to make up this part of the input file:

- the format of data collected by the data acquisition system. For some systems, this format is predefined within RAPID. For others, you must prepare inputs to define the format.

- how much (if any) of the input from the data collection you would like to have printed along with the output of RAPID. In addition, several options are available to determine how detailed the printed output of RAPID will be. This may range from a single table summarizing the test to the entire input data set, output data set, and plotted curves.

- title of the test — an up to 120 character description of the test. Usually, a test number and the date of the test will be included along with any desired description of the test. This title is printed at the top of most of the pages of the printed output of RAPID.

For this example test, the input lines are

```
    5    0    0    0    0    1    0    0
    4   -1    1    1    0    5
Furniture Calorimeter      Test 4
19 Oct 1989
```

which defines the following parameters (variable names, shown in italics, refer to names used in the detailed documentation in later sections of the document):

*INTYPE*      5      data acquisition system is a Hewlett-Packard[4] 9836)

[4] The use of company names or trade names within this report is made only for the purpose of identifying those hardware or software products with which the compatibility of the programs of RAPID has been tested. Such use does not constitute any endorsement of those products by the National Institute of Standards and Technology nor does it imply that those products are necessarily

| | | |
|---|---|---|
| *INPRT* | | 0 no printing of the data recorded by the data acquisition system. |
| *INPNCH* | 0 | data acquisition system output is not saved to disk once it has been analyzed. Note the original input data is still available. |
| *INSTOP* | 0 | continue program execution once the input data from the data acquisition system has been processed. |
| *INERR* | 0 | do not display individual errors discovered during processing of data acquisition system input. A summary of the number of errors is included in the printed output. |
| *INSKIP* | 1 | some input records from the data acquisition system are to be skipped during processing. This option is also used to identify the instrument calibration scans. |
| *INSAVE* | 0 | do not save the data acquisition system input in a data-acquisition-system-independent format. Since the processing of the data from the data acquisition system is sometimes the most time-consuming part of data analysis, it can be advantageous to translate the data to a format which can be efficiently read by RAPID. For small tests like this example, this is not necessary. |
| *INTEST* | 0 | keep any test number identification intact with incoming input data. This option only applies if INTYPE is equal to 2. |
| *NTEST* | 4 | identify this test as test number 4 on output saved to disk storage. |
| *NPRT* | -1 | print only a summary of the output dat |
| *NPNCH* | 1 | save the output data. Typically, this output file will be created by the program and assigned a unique name. Section 2.4 gives details of the default file naming conventions used by RAPID. |
| *NPLOT* | 1 | generate plotted output of select outputs. |
| *NCORR* | 0 | no corrections are to be made to the output data. |
| *NERR* | 5 | print a maximum of five warnings from each of the calculation commands. A summary of the total number of warnings for each calculation, if any, are included in the printed output. |
| *TITLE* | | Furniture Calorimeter Test 4 19 Oct 1989 |

### 5.2.2 Skipping Data Records on Input

The INSKIP parameter described above stipulated that some input records were to be skipped in the input data. This option allows you to delete incorrect or extraneous information produced by the data acquistion system. In addition, specifications for the automatic instrument calibrations are detailed. In this example, only the automatic instrument calibrations are specified.

```
SKIP=(Z2R4)6
```

This indicates that scans 1 and 2 as recorded by the data acquisition system are instrument zero calibrations and scans 3, 4, 5, and 6 are instrument full-scale range calibrations.

the most appropriate product for the implied use.

### 5.2.3 Test and Instrument Description

The next part of the control input (detailed in sec. 4.2) describes the instruments connected to the data acquisition system. Each instrument included in the input data set is described along with any linear conversions to be performed on sets of instruments. Each instrument is identified with a unique number (usually assigned by the data collection equipment), an instrument type (to identify the time channel and any linear conversion to be performed on the instrument), and a description of the instrument. The following information should be determined for this part of the input:

- the channel number of each instrument connected to the data acquisition system. Each instrument is identified by a unique number (called a *channel number*) when it is connected to a data acquisition system. In addition, time is assigned an arbitrary channel number (typically 0).

- the calibration of the instrument and any conversion (linear or nonlinear) necessary to convert the readings output by the data acquisition system into engineering units.

- an abbreviated and complete description of the instrument. The abbreviated description is printed at the top of all columns of printed output and is used as a legend for any graphical output.

For this example, these inputs are

```
00 1Time  Elapsed time                                             (s)
11 2CO2   CO2 Carbon Dioxide concentration                     (Vol %)
   0      Beckman CO2 analyzer, linear calibration
12 2CO2   Carbon Dioxide concentration, Beckman CO2 analyzer   (Vol %)
   0      serial # 34678, 10.0 % full scale, span gas 9.22%
13 3RADMTRRadiometer serial #624416, 30" away                  (kW/M²)
14 2CO    Carbon Monoxide concentration, Beckman CO analyzer   (Vol %)
   0      serial # 34753, 5.0% full scale, span gas 4.66%
15 2MARKEREvent marker                                             (V)
16 2WTLOSSTotal weight loss                                       (kg)
17 2O2    Oxygen concentration, linear calibration             (Vol %)
   0      Taylor O2 analyzer
18 2GASVELBi-directional velocity probe                          (m/s)
19 2SMOKE Smoke extinction coefficient                           (m-1)
20 2TC BOTTemperature at bottom of stack                          (°C)
21 2TC TOPTemperature at top of stack                             (°C)
22 2RMTEMPAmbient temperature at back of box                      (°C)
                                                                       999
 1.0
 1.0
 2635.5
                                                                       999
```

For two of these instruments, the inputs are detailed below.

*KH(1)* 0 Typically, channel number zero is used to identify the time channel. Any number not used by the data acquisition system may be used.

*ITYPE(1)* 1 The complete set of instruments are grouped into subsets for any linear conversions to be performed on the data. Instrument type 1 is *always* used to identify the time channel.

*Abbreviated name* TIME

*Full description* Elapsed time (s)

*KH(3)* 12

*ITYPE(3)* 2 Additional instrument types (in addition to the one for the time channel) may be defined as necessary to perform linear conversions. In this example, only types 2 and 3 are necessary since most instruments require nonlinear conversions (linear conversions for instruments of type 1 and 2 are simply a multiplication by 1.0).

*Abbreviated name* CO2

*Full description* Carbon Dioxide concentration, Beckman CO2 analyzer (Vol %) serial # 34678, 10.0 % full scale, span gas 9.22%

Note that more than one line of description can be included for an instrument. Usually information on the instrument (serial number, etc.) will be included along with any calibration information which the user would like stored with the test data.

### 5.2.4 Calculations Using Transformation Control Commands

The advantage of using RAPID for the analysis of fire test data lies in its extensive set of fire related calculations which may be used without any programming by the user. This technique provides a standard set of algorithms that can be applied to a variety of different testing scenarios. For this example, the calculations will range from simple transformation of voltage readings into engineering units to complex calculations using several instruments to determine the heat release rate of a burning item of furniture. The bulk of the document (secs. 6 through 9) describes the available commands in detail. Since the calculations for a test will vary depending upon the measurements taken and analysis desired, only a few of these calculations will be highlighted here. Note that a detailed description of each command is not provided in this section. Rather, the user is introduced to the types of commands available. Details of all of the commands are provided in sections 6 through 9.

Typically, the first step in a data analysis is the transformation of voltage readings into engineering units. For some instruments, this is a simple linear conversion. For others, a more complex calculation must be made. For this example, the input to make these conversions is shown below along with a description of the calculation performed.

```
TIME  60 second delay from data collection start to ignition
00 0 0 -60
```

Each transformation control command begins with a line identifying the command (`TIME`) and optionally a description of the calculation being performed. In this calculation, the time channel

(channel 0) is adjusted so that zero is the time of ignition by subtracting 60 seconds from each time reading.

```
GAS%  Convert gas analyzers from voltage to volume percent
4
11  3      0.    10.        5.    X
12  1 31   Z     R         93.5   X
14  1 32   Z     R         95.35  X
17  3      Z     0.834673  20.95  X
```

Calculations for several different types of gas analyzers are available with RAPID (see sec. 8.1 for details). Both a linear and a logarithmic calibration are shown in the four examples above (note that the spacing of the inputs is only for ease of use; spacing within a command is flexible). Details of the calculations for one of the analyzers is shown below.

*JCHAN* 12 Channel number 12 is a carbon dioxide analyzer.

*ITYPE* 1 It has a single range logarithmic calibration.

*IDNO* 31 A number of instrument calibrations are built into a configuration file used by RAPID. This analyzer is included in the catalog as number 31.

*ZERO* Z A "Z" indicates automatic determination of the output of the analyzer for zero gas concentration. As shown above, the instrument scans are tagged to show zero and span calibrations. These may optionally be used to allow RAPID to select the calibration endpoints automatically.

*SPAN* R An "R" indicates automatic determination of the output of the analyzer for a known concentration of gas.

*SM* 93.5 the output of the analyzer for the known concentration of gas. For this example, the output is 93.5 (on a scale of 0 to 100) for the gas concentration of 9.22 percent.

```
TC
1
20 22 X
```

Channels 20 through 22 are Chromel-Alumel thermocouples (the default thermocouple type).

```
VELOCITY
1
18  49.74   Z   3  1  21  X
```

| | | |
|---|---|---|
| *JCHAN* | 18 | Channel 18 is a bi-directional velocity probe. |
| *SEN* | 49.74 | The calibration of the probe is 49.74 Pa/Volt. |
| *ZERO* | Z | A "Z" indicates automatic determination of the probe output at zero velocity. |
| *JPTYPE* | 3 | A type 3 probe is a bi-directional velocity probe with positive change in output for positive gas velocity. |
| *JTMETH* | 1 | A single thermocouple located near the velocity probe is used to correct the gas velocity to STP. |
| *JTEMP* | 21 | Channel 21 is the thermocouple channel for temperature correction. |

```
DELAY  Time for gas to travel down exhaust pipe to measuring instruments
1
11  12  14  17  18  19  20  21  22  00  9.5 X
```

For some instruments, the measurement is made at a location remote from the experiment. The DELAY command allows you to account for the time delay and adjusts the readings (using a linear interpolation to calculate the values at the adjusted time scale). In this example, instruments 11, 12, 14, 17, 18, 19, 20, 21, and 22 are located 9.5 s away from the experiment (instrument number 00 is the time channel).

```
WT-LOSS Calculate total weight loss of specimen on load platform
1
16 192.6   X
SPECIFY  Re-use the weight loss channel for smoothed weight loss calculation
16 X
SMOOTH   Weight loss readings
1
16 00 7 X
```

| | | |
|---|---|---|
| *JCHAN* | 16 | Channel 16 is connected to a load cell to monitor specimen weight as the sample burns. |
| *CON* | 192.6 | is the conversion factor used to convert voltage to weight. |

In addition, once the weight loss is calculated, a running average smooth over 7 data points is calculated to reduce noise in the measurement. The SPECIFY command allows the results of a forthcoming calculation to be stored in an existing channel. For many of the commands, new columns of data are automatically created to hold the results of the calculation. In some cases, this results in unneeded data which may be eliminated with the SPECIFY command. The SMOOTH command calculates a running average smooth by fitting segments of the instrument readings to a linear equation. In this example, the weight loss (channel 16) and the time channel (channel 0) are used to calculate a 7 point running average smooth of the weight loss channel.

```
MASS-FLOW-3  Calculate mass flow of gas in exhaust stack for HRR calculation
18 0.18 21 .9 X
```

| | | |
|---|---|---|
| *JVEL* | 18 | the velocity channel |
| *AREA* | 0.18 | cross sectional area of the duct |
| *JTEMP* | 21 | gas temperature at the measurement point |
| *C* | 0.9 | empirical flow coefficient |

```
HEAT-RATE-2  Calculate heat release rate from stack measurements
17 $02 1 11  X
SPECIFY  Re-use the HRR channel for conversion from kW to MW
$03 X
COMPUTE  Change HRR from kW to MW
1
$03 / 1000. X
```

| | | |
|---|---|---|
| *JO2* | 17 | the oxygen concentration channel in volume percent. |
| *JMFR* | $02 | the calculated mass flow channel. Channels created by the program are automatically assigned pseudo-channel numbers beginning with $00, $01, etc. In this example, the mass flow calculation above created channel number $02. Since these numbers depend upon the quantity and type of calculations performed, a utility program is provided (CCLIST, sec. 2.5) to list the control input file with the created channels noted. |
| *ICO2* | 1 | correct the oxygen concentration to account for incomplete combustion using the carbon dioxide channel. |
| *JCO2* | 11 | the carbon dioxide channel. |

The calculated heat release rate is converted from kW to MW using one of the more powerful commands available. The COMPUTE command allows the user to enter any arbitrary calculation involving constants and one or more channels. In this example, a simple division by 1000. converts the created heat release rate channel (pseudo-channel number $03).

```
RENAME
    $1   SMOKE SMOKE EXTINCTION COEFFICIENT
    $2   MFLOW MASS FLOW RATE OF GAS THRU STACK (KG/S)
    $4   HRR   HEAT RELEASE RATE THRU STACK (MW)
                                                                              999
END
```

The RENAME command is use to provide an instrument list for created channels. The END command signals the end of the data transformation commands.

### 5.2.5 Plot Generation

In addition to printed output from the test, plots may be generated for selected output channels. See section 4.7 for details. An example is shown below.

```
PLOT 00 $4
Heat release rate measured in exhaust stack
0. 2500. 0. 1.00
Time - s                                        Heat release rate | MW
```

*IN* "PLOT 00 $04" specifies an X-Y plot of time (channel 00) versus the heat release rate channel (pseudo-channel $04).

*GTITL* "Heat release rate measured in exhaust stack" is printed at the top of the graph.

*XL,XH,YL,YH* minimum and maximum values for the X and Y axes are 0. to 2500. for the X and 0. to 1.00 for the Y.

*XTITL* "Time - s" is printed below the X axis as an axis legend.

*YTITL* "Heat release rate | MW" is printed vertically along the Y axis.

## 6 Transformation Control Commands

There are 34 commands that direct the program to perform the data transformations routinely required for fire test data. For discussion and documentation purposes only, these commands and subroutines can be divided into three classes: utility, basic, and complex. These are the Part H commands detailed in sections 7 through 9, below.

The utility class performs operations on reduced data such as integrating and averaging.

The basic class calculation is one in which, with the exception of temperature, only the values from one instrument are required. Typically, the basic class calculation does not create any new channels.

The complex class calculation requires at least two sets of instrument values or other information, such as instrument position. Typically, the complex class calculation creates one or more new channels in order to store the calculated results.

The subroutines that actually perform the data reduction are invoked by entering a command (beginning in column 1) with data input H1. When a subroutine is called, it will look for the additional input information needed to perform the data transformation. When the data transformation is complete, control returns to the main conversion subroutine, which looks for the next command to be executed. A brief description of each command is shown in the following sections.

### 6.1 Utility Commands (detailed in sec. 7)

| Class | Subpart | Command | Purpose |
|---|---|---|---|
| Utility | a | AMBIENTS | override default values of ambient temperature, pressure and relative humidity. |
| | b | AVERAGE | find the average of "n" channels; upper and/or lower limits may be set for each channel and the average may be weighted. |

|   |   |   |
|---|---|---|
| c | COMBINE | concatenate the values from more than one channel over specific intervals of the complete test in order to create a new, continuous, channel. |
| d | COMPUTE | find the result of any FORTRAN-like algebraic expression; operations are add, subtract, multiply, divide, raise to a power, find minimum or maximum; operands may be constants or channel numbers. |
| e | DELAY | adjust the values of specific channels to account for a delay in response, etc. |
| f | DELTA | find the difference between consecutive readings of the same channel. |
| g | E119 | create a channel with the standard E119 temperature (°C or °F) for each time scan using an identified time channel. |
| h | INTEGRATE | integrate a channel with respect to time. |
| i | RENAME | give meaningful names (other than the default names) to created channels. |
| j | RUNNING-SUM | create a channel that is the running sum of the data from some identified channel; very similar to INTEGRATE, but not done with respect to time. |
| k | SEPARATE | for channels that store information from more than one instrument, separate and store the individual results in individual channels. |
| l | SMOOTH | reduce the "noise" in a channel using a sliding least-squares straight line fit for small sections of the curve. |
| m | SPECIFY | specify the channel number for the following created channels. |
| n | STATS | calculate various statistics regarding any particular channel: minimum, maximum, average, time to exceed a particular value, etc. |
| o | TIME | convert h/m/s to elapsed s and/or add a time shift to the existing, or a new, time channel. |

### 6.2 **Basic Commands** (detailed in sec. 8)

| Class | Subpart | Command | Purpose |
|---|---|---|---|
| Basic | a | GAS% | calculate concentrations of different gases. |
| | b | PRESSURE | calculate static pressure. |
| | c | SMOKE | calculate smoke optical density. |
| | d | TC | convert voltage output to temperature for various different types of thermocouples. |
| | e | VELOCITY | calculate gas velocity. |
| | f | WT-LOSS | calculate total weight loss of monitored items. |

### 6.3 **Complex Commands** (detailed in sec. 9)

| | | | |
|---|---|---|---|
| Complex | a | BALANCE | calculate rate of heat release from total energy balance. |
| | b | FLOW-RATE | find neutral plane height and calculate volume flow rate, mass flow rate, and convective energy transport rate in and out of a chamber using gas velocity. |
| | c | GAS-FLOW | calculate the mass flow rate through an opening of any gas whose concentration, velocity, and temperature are known. |
| | d | HEAT-RATE | calculate the rate of heat release from gas concentration (oxygen depletion), gas velocity, and gas temperature. |
| | e | HEAT-RATE-2 | calculate the rate of heat release from oxygen depletion, gas velocity, and gas temperature; specifically designed for use when only one of each type of instrument is used (no profiles). |
| | f | HOT/COLD | find the position of the hot/cold interface as determined by the temperatures from an identified profile. |
| | g | MASS-FLOW | calculate neutral plane height and mass flow |

| | | |
|---|---|---|
| | | rate of gas through an opening using gas temperature profiles. |
| h | MASS-FLOW-2 | calculate the mass flow rate of gas through an opening using temperature profiles and a neutral plane height determined by another source. |
| i | MASS-FLOW-3 | calculate the mass flow rate of gas through an opening using a single gas velocity measurement and the area of the vent perpendicular to the gas flow. |
| j | STATIC | find the neutral plane height, thermal discontinuity height, pressure at the thermal discontinuity height, opening gas velocities, and interior temperatures from static pressure inside chamber. |
| k | SURFACE | calculate average and total heat loss rate and total incident heat flux to a surface using surface temperature. |
| l | VENT-LOSS | calculate radiative heat loss through an opening using exhaust gas temperature. |
| m | WT-RATE | calculate percent weight loss, rate of weight loss, and rate of heat release from total weight loss. |
| n | ZERO-TC | calculate zero diameter thermocouple temperatures from least squares fit of temperatures from various sized thermocouples. |

### 6.4 Data Input for Transformation Control Commands

The commands may be given in any order and as many times as necessary. The end of data transformation is signaled by entering the command "END." Note that many of the subroutines called "create" new channels in which to store the calculated or transformed results. These channels must be included when counting the number of channels used. Make sure the parameter NCOL in the main program, RAPID, is large enough. It is possible that up to 1000 channels may be created by the program, however, it is more typical that the same number of channels are created as are used (e.g., if 30 instruments are used in the experiment, around 30 channels will be created by the program). Of course, the number created also depends on the complexity of the data reduction being done.

All units are metric for both input and output unless otherwise noted.

**Input:** H1

**Variables:** *CMD, COMENT*
**Format:** A80

At least the *END* command must be entered! All commands MUST begin in column 1 and MUST NOT contain any spaces. However, there are some abbreviations that may be used. The end of the command is signified by at least one space; the rest of the input may contain any comment you wish to make. An unrecognizable command will cause program termination. The inputs required by the subroutine called should directly follow each command. When the transformation performed by the subroutine is complete, the next command may be entered. Note that any part of the command in brackets is optional and the commands may be in upper or lower case.

The available commands (and their acceptable abbreviations) are shown below. Each of these commands is explained in more detail in the following sections.

*A[MBIENTS], AV[ERAGE], B[ALANCE], COMB[INE], C[OMPUTE], DELA[Y], D[ELTA], E[119], F[LOW-RATE], G[AS%], GAS-[FLOW], H[EAT-RATE], H[EAT-RATE-]2, HO[T/COLD], I[NTEGRATE], M[ASS-FLOW], M[ASS-FLOW-]2, M[ASS-FLOW-]3, P[RESSURE], R[ENAME], RU[NNING-SUM], S[EPARATE], SM[OKE], SMOO[TH], SP[ECIFY], ST[ATIC], STATS, SU[RFACE], TH[ERMOCOUPLE] or TC, T[IME], V[ELOCITY], VEN[T-LOSS], W[T-LOSS], WT-R[ATE], Z[ERO-TC].*

The *END* command signals the end of the calculations.

*COMENT* can be used to describe the calculation. Beginning with the first position after the space, any comment can be made.

At this point enter the input specified under each command as it is given.

## 7 Control Input for Utility Commands

The utility commands perform operations on reduced data such as integrating and averaging.

### 7.1 AMBIENTS: Ambient Conditions

The command can assign values other than default values to the ambient temperature (°C), pressure (kPa), and relative humidity (%). The ambient air density (kg/m$^3$) is calculated using the ambient temperature and pressure and is not available to be set.

The default ambient values are:

| | |
|---|---|
| Temperature | = 20 °C |
| Pressure | = 101.3 kPa |
| Relative Humidity | = 50. % |
| Air Density | = 1.205 kg/m$^3$ |

It is possible to give the command AMBIENTS more than once. However, any ambient that is set using this command, will remain at the value given until reset also using this command. The values set here are in effect and are available throughout the program. Any other command that is called that requires an ambient value will use the most current value of the ambient. It is not necessary to issue the command AMBIENTS if the default values are to be used.

**Input:** HUa1

**Variable:** *CTRL*
**Units:** Temperature - °C, Pressure - kPa, Relative humidity, %RH
**Format:** A80

Only one of these inputs is read each time the command AMBIENTS is given. The program searches for each of the three "key-words" that identify which ambient value is to be set:

“AMBT=” - set ambient temperature
“AMBP=“ - set ambient pressure
“AMBRH=“ - set ambient relative humidity

After each keyword, the next five characters are assumed to be the value of the ambient being set, in F format (e.g. “AMBT= 238.54” would assign the value 238. °C to the ambient temperature; the 5 and 4 are ignored since they are the sixth and seventh characters. Any or all of the ambient values may be set with this control input. If more than one value is set, be sure that at least 5 characters (including blanks) separate each key-word. The key-words may appear on the control input in any order.

*Example*

```
AMBIENTS
AMBT=20.     AMBRH=43.   AMBP=110.3
```

### 7.2 AVERAGE: Weighted Averages

The command finds the average of the values from up to 25 channels. The average may be a weighted average and lower and/or upper limits for the values may be specified.

The general format for entering the information for the average is as follows:

[<limit$_1$>] channel$_1$ [weight$_1$] [<limit$_2$>] channel$_2$ [weight$_2$] ... [<limit$_i$>] channel$_i$ [weight$_i$] X
for 1 <= i <= 25.

In the format above, brackets (i.e., []) surround values that are optional inputs (the brackets themselves should not appear on the input). Note that the limits (limit$_i$) appear within "limit delimiters" (<>) and that, once set, apply to the values of all channels (channel$_i$) that follow until they are reset or changed. You may specify a lower limit only, an upper limit only, both a lower and an upper limit, or no limits. Initially no values are set and, thus, no limits will be used unless specified. Conversely, the weighting factor (weight$_i$) only applies to the channel that it follows. If no weighting factor is specified, the weighting factor defaults to 1.0. The commas are optional. The "X" signals the end of the input and is strictly required. Often, one line will not allow sufficient space to enter all the input desired. Therefore, the input may appear on any number of lines as long as not more than 500 characters are used (including spaces and the "X").

The specific format for entering limits is as follows:

$$\left( < \left( \left( \begin{pmatrix} \text{R} \\ \text{or} \\ \text{D} \end{pmatrix} \text{lower limit} \right) \left( , \left( \begin{pmatrix} \text{R} \\ \text{or} \\ \text{D} \end{pmatrix} \text{upper limit} \right) \right) \right) > \right).$$

As in the general format above, values or characters within a set of brackets are optional. To identify that limits are being entered, a set of less-than (<) and greater-than (>) symbols (limit delimiters) must surround the limits. The limits themselves may be real or integer values; however, an integer value is assumed to be a channel number. If the limit is a real value, the value of the limit is constant. If the limit is a channel number, the value of the limit is the value stored in the channel at the time being averaged. If a channel number that does not exist is entered as a limit, the run terminates.

The "R" and "D" stand for REPLACE and DELETE, respectively. If a value is outside a limit that has an "R" attached to it, the value being averaged is replaced by the limit value. If outside a limit that has a "D" attached, the value is not used to find the average and the population of the average is decreased by the weighting factor for that channel. If neither "R" nor "D" is entered, "R" is assumed.

A lower limit only, an upper limit only, both lower and upper limits, and no limits may be specified. The different combinations of limits is obtained as follows (—— indicates a channel or constant value):

| | |
|---|---|
| <——> or <——,> | set lower limit only, no upper limit |
| <,——> | set upper limit only, no lower limit |
| <——,——> | set both limits |
| <> or <,> | set no limits |

Note that an empty field is not the same as having a limit set to zero. Also note that a comma may appear if a lower limit is set but must appear if an upper limit is to be set (an error will occur if one is not found).

The specific format for entering a weighting factor is as follows:

[*——]

where —— stands for the weighting factor value.

The brackets again indicate that the weighting factor is an optional entry. To identify that a weighting factor is being entered, a "*" must be prefixed to the weighting factor value. The value itself may be a real or an integer, but will be treated by the program as a real.

Note that anywhere in this command that a channel number is allowed as an entry, a created channel number can be specified by using the method described in the Notes on the Preparation of Data Inputs Read by RAPID (see NPDI [3]).

Following is an example of a typical input record:

321 322*1.5 323 <320,D1000.> 324*2 325 <> 326 <,R1000.> 327 X.

The record above would be deciphered as follows:

| Channel | Weight Factor | Lower Limit | Channel or Constant | Replace or Delete | Upper Limit | Channel or Constant | Replace or Delete |
|---|---|---|---|---|---|---|---|
| 321 | 1.0 | none | - | - | - | - | - |
| 322 | 1.5 | none | - | - | - | - | - |
| 323 | 1.0 | none | - | - | - | - | - |
| 324 | 2.0 | 320 | channel | replace | 1000. | constant | delete |
| 325 | 1.0 | 320 | channel | replace | 1000. | constant | delete |
| 326 | 1.0 | none | - | - | none | - | - |
| 327 | 1.0 | none | - | - | 1000. | constant | replace |

One channel is created by the program for each average found.

**Input:** Hub1

**Variable:** *NAVG*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command AVERAGE is given.

*NAVG* is the number of average calculations to be made. Prepare NAVG sets of HUb2 inputs. NAVG channels will be created.

**Input:** Hub2

**Variable:** *IN*
**Format:** A80

Any number of these inputs may be used for a single average (up to 500 characters including spaces and the "X").

*IN* is the input record as described in the discussion above.

*Example*

```
AVERAGE  STACK TEMPERATURE AND GAS VELOCITY
2
61 62 63 64 65 66 67 68 69 X
51 52 53 54 55 56 57 58 59 X
```

### 7.3 COMBINE: Combining Multiple Channels

The command concatenates values from identified channels over specified ranges. The resulting data vector is stored in a channel created by the program. It is useful for combining into one channel, the values from two or more channels, such as when two instruments with different ranges are used to measure the same phenomenon.

A time (in seconds) or a scan number may be used to identify when the values from a channel are to begin being included. If a time is used, the actual scan number (j) is determined internally such that t(j-1) < time <= t(j), and where t(i) is the time at the ith scan. The values from a channel continue to be included until the next beginning scan number is reached (if there is one). If desired, times and scan numbers can both be used (a time for one channel, a scan for the next, etc.).

If the range of the first channel begins sometime after the first scan, the scans of the created channel are undefined and arbitrarily set to zero.

The scan numbers (or times) need not be entered in any particular order except that they should match the same relative order in which the channel numbers to be combined were entered. Ordering of the scans is done internally.

For example, if the combined channel were to be made up of three different data channels over four intervals, the inputs might look like this:

```
621  622  621  650  X
0.  300.  600.  250  X
```

This set of inputs instructs the command to create a channel made up of the values from channel 621 from time 0. up to, but not including, the value at time 300. seconds. From 300. up to, but not including, 600., the values from channel 622. From 600. seconds up to, but not including, the value at scan 250, the values from channel 621 again. And from scan 250 to the end, the values from channel 650.

**Input:** HUc1

**Variables:** *NCOMB [, JTIME]*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command COMBINE is given.

*NCOMB* is the number of combinations of channels to be made. Prepare NCOMB sets of HUc2 and HUc3 inputs. NCOMB channels will be created.
*JTIME* is the time channel number. If times are to be used to define the beginning of an interval, this value must be entered.
*X* = ‘ X’ - end-of-set mark.

**Input:** HUc2

**Variables:** *JCHAN(i) X* *1 <= i <= 20*
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the channel number of values to be included in the combination (NPDI [3]).
*X* = ‘ X’ - end-of-set mark.

**Input:** HUc3

**Variables:** *(IB(i) or BTIME(i)) X* *1 <= i <= 20*
**Format:** EVALU8 (NPDI [2])

These values are used to identify the beginning of the interval corresponding to the data channels above.

*IB* is a scan number.
*BTIME* is a time in seconds. BTIME may not be used if JTIME is not specified above.
*X* = ' X' - end-of-set mark.

*Example*

```
COMBINE
1 0 X
23 24 25 X
0. 300. 910. X
```

### 7.4 COMPUTE: User Specified Calculations

The command deciphers a FORTRAN-type algebraic expression and calculates the results. The operations that can be handled are ADD (+), SUBTRACT (-), MULTIPLY (*), DIVIDE (/), RAISE TO A POWER (**), AVERAGE (A), FIND THE MINIMUM (<), and FIND THE MAXIMUM (>). (Note that AVERAGE (A), MINIMUM (<), and MAXIMUM (>) have non-standard operator symbols). The operators and operands may be nested in parentheses in order to perform the operations in the desired sequence. The operators themselves have hierarchical ranks as follows:

| Function | Symbol | Rank |
|---|---|---|
| AVERAGE | A | 3 |
| RAISE TO A POWER | ** | 2 |
| MULTIPLY or DIVIDE | * or / | 1 |
| ADD or SUBTRACT | + or - | 0 |
| MINIMUM or MAXIMUM | < or > | 0 |

The AVERAGE function generates the appropriate ADDs, DIVIDEs, and nesting to insure the proper average is found. Note that, unlike the averaging algorithm employed when the command AVERAGE is used, no limits or weights can be used for this average. When several AVERAGE operators are encountered in a string (unbroken by parentheses or other operators) all the values linked together by the average symbols (A) are added before the average is found.

The MINIMUM operation finds and saves the smaller value of two values and the MAXIMUM operation does the same only for the larger value.

The operands may be either real constants or channel numbers. If an operand contains a decimal point, it is assumed to be a constant. Otherwise, the operand is assumed to be a channel number. If you wish to use channels which were created by the program, you may do so by using the method described in the Notes on the Preparation of Data Inputs Read by RAPID (see NPDI [3]). If an assumed channel is not found, the run will terminate.

A typical computation might be as follows:

```
(301 A 302 A 303 * ((20.9 - (313 < 314)) / 100.)) ** 2.  X .
```

In the example above, the first step is to find the smaller value from channels 313 and 314 and then subtract that value from the constant 20.9. That result is then divided by 100.. The average of the values from channels 301, 302, and 303 is then found and then multiplied by the result of the above division. Finally, that result is squared. The "X" indicates the end of the computation and must be present. Up to 500 significant (non-blank) characters may be used to enter one computation (including the "X").

In addition there are three so-called "channel operators." The three channel operators are H (for HIGH), L (for LOW), and M (for MEAN). These operators are used to find a single value within a single channel. The syntax is to use the operator (H, L, or M) followed by a channel number; e.g., H408, L16162, M$03. The operator and its channel number are reduced to a single real number before any other calculation is done. Thus the channel operator/channel number may be used anywhere a real number may be used and must follow any syntax pertaining to real numbers.

| | |
|---|---|
| H <channel number> | returns the highest value found in the channel. |
| L <channel number> | returns the lowest value found in the channel. |
| M <channel number> | returns the average of all the values found in the channel. |

One channel is created by the program for each computation performed.

**Input:** HUd1

**Variable:** *NCOMP*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command COMPUTE is given.

*NCOMP* is the number of computations to be done. Prepare NCOMP sets of HUd2 inputs. NCOMP channels will be created.

**Input:** HUd2

**Variable:** *IN*
**Format:** A80

As many inputs as needed may be used but the total number of significant (non-blank) characters may not exceed 500.

*IN* is the input computation as described in the discussion above.

*Example*

```
COMPUTE  MULTIPLY HRR BY EMPIRICAL COEFFICIENT AND CONVERT FROM KW TO MW
3
$7 * 0.7 / 1000. X
(750 -L750) * 4.66 / 1000. X
0.002552+0.007788*380*1000.+2.4*10.**(-5.)*(380*1000.)**2.+2.*10.**(-8.)*
(380*1000.)**3. X
```

### 7.5 DELAY: Instrument Response Time Delay

This command accounts for any delay in the output due to the response time of an instrument. For any response time, r, a reading, R, at time, t, is defined as: R(t) = R(t+r). If the response time is not an even multiple of the scan rate, a straight line interpolation of the data is done.

For time, x < t+r < y, the interpolation and redefinition is:

$$R_t = R_{t+r} = \left(\frac{(t+r)-x}{y-x}\right)\times\left(R_y - R_x\right) .$$

No new channels are created by this command. Any changes in the data matrix take place in the identified channel.

**Input:** HUe1

**Variable:** *NDLAY*
**Units:** time - s
**Format:** open (NPDI [1])

NDLAY is the number of groups of channels for which delays are to be entered. Enter NDLAY HUe2 inputs.

**Input:** HUe2

**Variables:** *JCHAN(1), JCHAN(2), . . . , JCHAN(i), JTIME, DLAY X* *1 <= i <= 98*
**Format:** EVALU8 (NPDI [3])

*JCHAN* is the channel number of the values which are delayed (NPDI [3]).
*JTIME* is the time channel number.
*DLAY* is the amount of the delay. Note that the units for the amount of the delay must be the same as the units of the time channel.
*X* = ' X' - end-of-set mark.

*Example*

```
DELAY
1
187 188 189 23 64 0 9.5 X
```

### 7.6 DELTA: Calculating Sequential Differences

The command calculates the difference between sequential values of any particular channel. The results are stored in a channel created by the program. Any channel number may be used as input. If you wish to use a channel which was created by the program, you may do so by using the method described in the Notes on the Preparation of Data Inputs Read by RAPID (see NPDI [3]).

There are two methods of calculation and storage of results as follows:

Method 1:
$D(1) = 0.$
$D(i) = r(i) - r(i-1)$, *for* $2 <= i <= N$

Method 2:
$D(i) = r(i+1) - r(i)$, *for* $1 <= i <= N-1$
$D(N) = 0.$

where D is the resulting difference, r is the value stored in the channel, and N is the total number of scans.

**Input:** HUf1

**Variables:** *NCHAN*
**Format:** open (NPDI [1])

*NCHAN* is the number of channels for which the incremental differences between adjacent values are calculated. Prepare NCHAN HUf2 inputs. NCHAN channels are created.

**Input:** HUf2

**Variables:** *JCHAN, [ICALC] X*
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the channel number containing the values between which the differences are calculated (NPDI [3]).
If *ICALC* is equal to zero, use method 1, above. If *ICALC* is not equal to zero, use method 2, above. Note that the default is method 1.
*X* = ' X' - end-of-set mark.

*Example*

```
DELTA
1
121 X
```

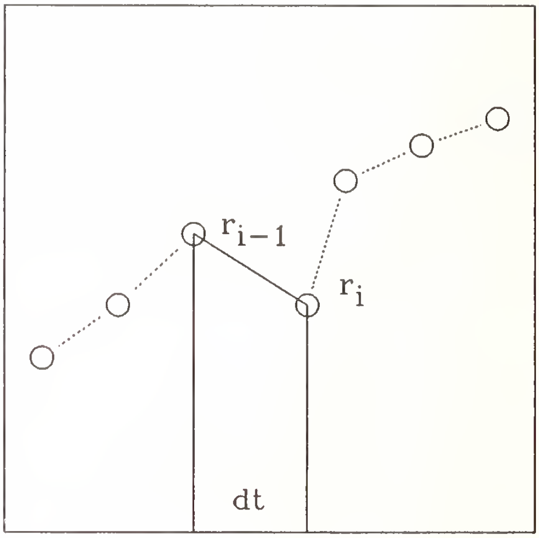
### 7.7 INTEGRATE: Numerical Integration

The command calculates the area under the curve with respect to time. The area is calculated using a geometric algorithm (the sum of the trapezoids defined by the segment of the curve between two points and the time axis):

$$A_i = \sum_{i=1}^{n} a_i = \sum_{i=1}^{n} \frac{1}{2}(r_{i-1}+r_i)dt$$

where $A_i$ is the total area under the curve up to point i and $a_i$ is the area of the "ith" trapezoid.

The calculated values are stored in a channel created by the program. Note that the values being integrated have already been transformed using the conversion coefficients input on Data Input B3 for the original channel. The conversion coefficients for the created channels are the default values of C = 1.0, ADD = 0.0, and POWER = 1.0.



**Input:** HUg1

**Variables:** *NINT*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command INTEGRATE is given.

*NINT* is the number of curves to be integrated. Prepare NINT HUg2 inputs. NINT channels will be created.

**Input:** HUg2

**Variables:** *JCHAN, JTIME X*
**Units:** time - s
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the channel number of curve to be integrated (NPDI [3]).
*JTIME* is the reference time channel number with respect to which JCHAN is integrated.
*X* = ' X' - end-of-set mark.

*Example*

```
INTEGRATE  CALCULATE TOTAL HR (MJ) FROM RATES OF HEAT RELEASE
2
$8 000 X
$9 000 X
```

### 7.8 RENAME: Descriptions for Created Channels

The command allows the user to provide descriptions of the channels created by the program other than the descriptions generated by the program when a channel is created. The format used is the same as that for data inputs D1 and D2 except that the "$" signifying a created channel is recognized and the instrument type need not be identified.

**Input:** HUh1

**Variables:** *CHAN, ID1, ID2, KHPRT*
**Format:** A6, 2X, A6, A60, A3

The number of HUi1 inputs is variable. The end of HUi1 format inputs is signaled by entering one HUi2 input.

*CHAN* is the channel number being identified, right justified and including the "$" (if desired) in the left-most significant position.
*ID1* is a six-character, abbreviated identification.
*ID2* is an up to 60 character description.
If ***KHPRT*** is blank, print out the values of this column on output. If ***KHPRT*** is nonblank, do not print out the values of this column. This can be used to shorten the printed output by not including intermediate calculations in the printed output.

**Input:** HUh2

**Variables:** *IEND*
**Format:** 77X, A3

*IEND* equal to '999' signals the end of the set of HUh1 inputs.

*Example*

```
RENAME
   380  CO   TBCO   CONC TOP BOX, BECKMAN RANGE 3, SP%=1.00    (VOL %)
   381  CO2  TBCO2  CONC TOP BOX, BECKMAN RANGE 2, SP%=4.99    (VOL %)
   $13                                                                    *
   $14  COMASSMASS FLOW RATE OF CO THROUGH STACK (G/S)
   $15  NPH BRNEUTRAL PLANE HEIGHT IN BURN ROOM DOOR (M)
   $16  MFI BRMASS FLOW RATE IN  THROUGH BURN ROOM DOOR (G/S)
   $47  COTOT TOTAL MASS OF CO THROUGH STACK (KG)
                                                                            999
```

### 7.9 SEPARATE: Separating One Channel Into Several Channels

At times one channel will be used to collect the data from more than one instrument by manually or automatically stepping a switch, etc. In order to plot, or otherwise compare, the individual values, it is necessary to resolve the channel into its separate parts. Since the time at which the "nth" value of each part is recorded is different, the time channel must also be resolved into separate parts.

The command resolves both the data and time channels into parts according to a pattern, or "separation code," provided by the user. Various, repeatable patterns may be needed to identify which scans belong to which instrument. The separation code format allows for an infinite number of variations.

In general, the separation code consists of the "skip/keep" pattern in parentheses followed by the scan number of the last scan to which the pattern applies:

([skip/keep pattern]) [scan number] ([skip/keep pattern]) [scan number] ...

The skip/keep pattern is repeated until the scan number is reached. The pattern itself consists of a series of up to 80 letters and numbers signifying skip or keep and how many scans of each. The letter 'S' stands for skip; the letter 'K' stands for keep. Any number of alternating skips and keeps may appear in the pattern.

Example 1

The channel has the values from three instruments stored in it and each instrument is recorded every third scan. There were 295 scans taken. The separation codes for the three instruments would be:

| | | |
|---|---|---|
| Instrument 1: | (K1S2)295 | (keep scans 1, 4, 7, ..., 295) |
| Instrument 2: | (S1K1S1)295 | (keep scans 2, 5, 8, ..., 293) |
| Instrument 3: | (S2K1)295 | (keep scans 3, 6, 9, ..., 294) |

Example 2

One set of instruments' values stored in the channel was recorded such that, to recover them, every first, third, fourth and sixth scans are skipped and every second and fifth are kept, up to scan 100. From scan 101 to 295 (end of test), every other scan is kept. The separation code would be:

(S1 K1 S2 K1 S1)100 (S1 K1)295

Alternatively, the letter 'E' may be used to signify the last scan recorded:

(S1 K1 S2 K1 S1)100 (S1 K1)E

Spaces in the code are ignored except that if a space follows a number, the entire number is assumed to be to the left of the space:

```
(S1K1S   2 K1 S1)  100(S 1K 1  )  E
```

is the same as the code above.

The following syntax rules should be used when preparing the separation code:

1. A '(' should be the first significant character on the input and should be the first significant character, if any, after a scan number.
2. An 'S' or a 'K' must be the next significant character after '('.
3 A "repeat" number must follow 'S' or 'K' even if it is 1.
4. An 'S', 'K', or ')' must be the next significant character following a repeat number.
5. A scan number or 'E' must be the next significant character(s) following ')'.
6. Spaces are ignored except that if a space follows a number, the whole number is assumed to be to the left of the space.

Note that if an 'E' is not used and the last scan number given is not greater than or equal to the number of scans in the channel, all scans past the last scan number given are skipped. If the last scan number given is greater than the number of scans used, the "skip/keep" pattern is repeated until the last scan used is reached.

The conversion coefficients of the original channel (identified by ITYPE on data input D1 and entered with data input D3) are also used by the newly created channel. Both the separated values and the separated time are stored in new channels created by the program.

**Input:** HUi1

**Variables:** *JTIME, JCHAN, NPART X*
**Units:** time - s
**Format:** EVALU8 (NPDI [2])

Only one of these inputs is read each time the command SEPARATE is given.

*JCHAN* is the channel number of data to be separated into parts (NPDI [3]).
*JTIME* is the reference time channel number.
*NPART* is the number of parts into which data is to separated. Prepare NPART HUk2 input(s).
*X* = ' X' - end-of-set mark.

**Input:** HUi2

**Variables:** *PAT*
**Format:** A80

Two channels are created for every HUk2 input read: the first is for data, the second is for time.

*PAT* is the 80 character separation code as described above. The following rules apply:

1) first character must be ‘(’.
2) ‘S’ or ‘K’ must follow ‘(’.
3) “repeat” number must follow ‘S’ or ‘K’.
4) ‘S’, ‘K’, or ‘)’ must follow “repeat” number.
5) scan number or ‘E’ must follow ‘)’.
6) spaces are ignored except they must not be in the middle of numbers.

*Example*

```
SEPARATE
123 0 3 X
(S1 K1 S2 K1 S1)100 (S1 K1)295
(K1 S1 K2 S1 K1)100 (K1 S1)295
(S1)295 (K1)E
```

### 7.10 SMOOTH: Data Smoothing

The command will smooth “noisy” data by finding the least squares straight line fit for small sections, or “windows,” of the curve. The window is an odd number of adjacent values from the channel being smoothed along with the corresponding values from the time channel. The smoothed value at the center of the window is calculated from the equation of the fit line and stored in a new channel created by the program. After the smoothed point is calculated and stored, the window slides up, dropping the “oldest” value and adding a new one. A new straight line is determined and the point calculated and stored. The process is repeated until the entire curve is smoothed.

Note that any odd number of points may be used, but a small number is preferable when the trend of the curve shows a rapid increase or decrease.

It is possible to multiply-smooth a curve simply by entering the created channel number of a previously smoothed curve as the channel to be smoothed (see NPDI [3]).

The equation of the line is of the form: $S_i = A * x_i + B$, where $S_i$ is the calculated “$i^{th}$” smooth point, $x_i$ is the “$i^{th}$” time value, and A and B are determined by:

$$A = \frac{n\sum_{i=a}^{b} x_i y_i - \sum_{i=a}^{b} x_i \sum_{i=a}^{b} y_i}{n\sum_{i=a}^{b} x_i^2 - \left(\sum_{i=a}^{b} x_i\right)^2}$$

$$B = \frac{\sum_{i=a}^{b} y_i - A\sum_{i=a}^{b} x_i}{n}$$

where n is the number of points in the window between point a and point b, $x_i$ is the "$i^{th}$" time reading, and $y_i$ is the "$i^{th}$" value from the curve being smoothed.

Note that the values being smoothed have already been transformed using the conversion coefficients input on data input D3 for the original channel. The conversion coefficients for the created channels are the default values of C = 1.0, ADD = 0.0, and POWER = 1.0.

**Input:** HUj1

**Variables:** *NCHAN*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command SMOOTH is given.

*NCHAN* is the number of channels to be smoothed. Prepare NCHAN HUl2 inputs. NCHAN channels will be created.

**Input:** HUj2

**Variables:** *JCHAN, JTIME [, NPTS] X*
**Units:** time - s
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the channel number of data to be smoothed (NPDI [3]).
*JTIME* is the time channel number.
*NPTS* is the number of values in the running smooth "window" (default value is 3).
*X* = ' X' - end-of-set mark.

*Example*

```
SMOOTH   Weight loss readings
1
16 00 7 X
```

### 7.11 SPECIFY: Specifying Created Channel Numbers

The command allows the user to specify the channel number that the next "n" created channels will be given (see NPDI [3]). Up to 100 channel numbers can be specified at a time. A channel number can be specified more than once with no message given. If a specified channel has already been used, the values in that channel are over-written. If all the channels from a previous SPECIFY command have not been used when a new SPECIFY command is given, the left-over channels from the previous command are NOT used. It is possible for more than one created channel reference ($xxx) to point to the same channel and a single channel may, at different points in the execution, contain more than one set of reduced data. It is therefore very important that care be exercised when using this command.

*Note that this can be a very dangerous command. Care must be taken to insure that the correct channel is used for the calculations which follow the specify command.*

**Input:** HUk1

**Variables:** *JSPEC(i) X* *1 <= i <= 100*
**Format:** EVALU8 (NPDI [2])

*JSPEC* is a channel number to be used when a channel is created, in order of usage (NPDI [3]).
*X* = ' X' - end-of-set mark.

*Example*

```
SPECIFY  Re-use the weight loss channel for smoothed weight loss calculation
16 X

SPECIFY  Re-use the HRR channel for conversion from kW to MW
$03 X
```

### 7.12 STATS: Interval Statistics

The command calculates various statistics with respect to the values from any particular channel. It finds the minimum and maximum values and the times at which those values first occur. It calculates the average value for the test. It compares the values from the channel to a constant or the values from some other channel and determines:

1. the first time at which the value of the channel is less than the comparison value and greater than the comparison value, and

2. the total time less than the comparison value, and greater than the comparison value.

3. In addition, the time range over which the statistics are determined can be specified.

This command does not reduce any raw data and does not create any new channels.

**Input:** HUl1

**Variables:** *NSTATS*
**Format:** open (NPDI [1])

*NSTATS* is the number of blocks of channels for which statistics are to be determined. Prepare NSTATS HUl2 inputs.

**Input:** HUl2

**Variables:** *JCHANL, JCHANH, CMPVAL or JCOMP, JTIME [, TIMELO or ITIMEL [,TIMEHI or ITIMEH]] X*
**Units:** time - s
**Format:** EVALU8 (NPDI [2])

*JCHANL* is the first channel number in the block of channels (NPDI [3]).
*JCHANH* is the last channel number in the block of channels (NPDI [3]). Note that JCHANL and JCHANH should be determined by the order of the instrument list (data inputs D1). If there is only one channel in the block, JCHANH should be the same as JCHANL which should be the channel.
*CMPVAL* is a constant comparison value.
*JCOMP* is the channel number of the values to be used for comparison.
*JTIME* is the time channel number.
*TIMELO* is the time(s) at which to begin determining the statistics; default is time of first scan.
*ITIMEL* is a scan number at which to begin determining the statistics; default is 1.
*TIMEHI* is the time(s) at which to end determination of statistics; default is time of last scan.
*ITIMEH* is a scan number at which to end determination of statistics; default is the last scan.
*X* = ‘ X’ - end-of-set mark.

*Example*

```
STATS  BOTTOM BOX
3
35 37 0. 000 180. 1980. X
41 43 0. 000 240. 2040. X
38 40 0. 000 495. 2295. X
```

### 7.13 TIME: Converting and Shifting Time Readings

The command can change the readings stored in a specified channel from hours/minutes/seconds format to elapsed seconds, and/or add a time shift. The results may be stored back in the original channel (destroying the old values) or they may be stored in a new channel (saving the old values in the original channel).

**H/M/S Conversion to S:** The hours / minutes / seconds to seconds conversion is not needed unless the program expects the time reading to be in seconds format but was recorded in hours / minutes / seconds format. If the program gets the format it expects, the time is automatically converted to seconds.

**Time Shift:** The time shift is used when "time-zero" occurs before or after the data acquisition system is started. If the event marking the beginning of the test occurs before the start of the data acquisition, the time shift will be a positive value; if it occurs after, the time shift will be a negative value.

**New Channel:** The new channel is created automatically by the program at the option of the user. However, it is not always necessary to store the adjusted time in a new channel. If the old values in the original channel are not required, and the size of the data matrix is critical, it is recommended that a new channel not be used.

Note that, alternatively, if a new channel is not needed, the hours / minutes / seconds to seconds conversion and the time shift can be performed without using the TIME command by using appropriate coded part D (instrument identification and conversion coefficient) and part E (reading format) data inputs.

**Input:** HUm1

**Variables:** *JTIMEO, ITIME, IHMS, TSHIFT*
**Units:** time - hh:mm:ss or s
**Format:** OPEN (NPDI [1])

Only one of these inputs is read each ime the command TIME is given.

*JTIMEO* is the original time channel number.
If *ITIME* is greater than zero, create new time channel to store adjusted time (1 channel created).
If *IHMS* is greater than zero, perform h/m/s to s conversion.
*TSHIFT* - time shift, in the same units as the original time channel.

*Example*

```
TIME  60 SECONDS TO IGNITION
0  0  0  -60
```

## 8 Control Input for Basic Commands

The basic class calculation is one in which, with the exception of temperature, only the values from one instrument are required. Typically, the basic class calculation does not create any new channels.

### 8.1 GAS%: Gas Analyzers

The subroutine calculates the volume percent concentrations of gas from the output of four different types of analyzers.

**Types 1 and 2**

For analyzer types 1 and 2, the concentration is calculated using a natural log fit of the calibration curve (actually the inverse of Beer's Law, which is an exponential). The calibration curve is the relationship of the concentration to the analyzer meter reading and not necessarily to the recorded analyzer output. The actual calculation is made by first changing the recorded output to an equivalent meter reading and then using that meter reading in the calibration curve equation to find the concentration:

$$M = \frac{R-R_0}{R_s-R_0} \times M_s$$

Then,

$$C = -a \ln\left(1-\frac{M}{b}\right) .$$

Type 2 analyzers differ from type 1 analyzers only in that they may use two ranges during a test and, thus, two calibration curves. The range change may be indicated automatically by a voltage change in a dummy channel or it may be indicated by entering the scan number or time of the switch. There are advantages and disadvantages to each type of indicator. The automatic indicator allows an unlimited number of switches between the two ranges but the output voltages must be less than and greater than one volt to indicate the change. If either the scan number or the time is used to indicate the switch, only one switch can be used. In addition, if the time is used, the time channel number must have an ITYPE equal to 1 (see Data Input D1).

Type 3

The change in output from type 3 analyzers is linearly proportional to the change in concentration of the gas. Once the slope of the line is found, any concentration may be calculated. To define the slope, two points must be known, typically the "zero" and one other point. The concentration, $C$, for any output $R$ then becomes:

$$C = \frac{C_s(R-R_0)}{R_s-R_0} .$$

Type 4

The concentration recorded by type 4 analyzers is proportional to $10^{(-R/k)}$.

To find $k$, two concentrations and their corresponding outputs must be known:

$$k = \frac{R_2-R_1}{\log(C_1/C_2)} .$$

Note that neither $C_1$ nor $C_2$ may be zero.

Then the concentration, $C$, for any output $R$ is

$$C = \frac{C_s}{10^{(R-R_s)/k}} .$$

For all four types of analyzers, the calculated concentrations (volume percent) replace the raw data values in the data matrix; no new channels are created.

**Input:** HBa1

**Variables:** *NGAS*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command GAS% is given.

*NGAS* is the number of gas analyzers. Prepare NGAS set(s) of HBa2 through HBa5 inputs.

**Input:** HBa2.1

**Variables:** *JCHAN, ITYPE, IDNO, ZERO, SPAN, SM [, CA, CB, SERNO, GAS, RANGE] X*
**Units:** gas concentration - output units of the analyzer, typically volume %
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the analyzer channel number (NPDI [3]).
*ITYPE* is the analyzer type code. Use this input format only if ITYPE = 1, single range calibration curve.
*IDNO* is the curve number of analyzer and range to be used. If analyzer is not in catalog, enter 0 and enter CA, CB, SERNO, and GAS below.
*ZERO* is the output for zero concentration of gas. If the “zero” scans have been identified in Part F, the character “Z” may be used in place of a number.
*SPAN* is the output for a known concentration of gas. If the “span” scans have been identified in Part F, the character “R” may be used in place of a number.
*SM* is the meter reading for span concentration.
*CA, CB* are the calibration curve coefficients. Enter these values only if IDNO=0.
*SERNO* is the analyzer serial number, enclosed in quotation marks. Enter these values only if IDNO=0.
*GAS* is the type of gas analyzed, enclosed in quotation marks. Enter these values only if IDNO=0.
*RANGE* is the maximum concentration of gas for the analyzer in quotation marks. Enter these values only if IDNO=0.
*X* = ‘ X’ - end-of-set mark.

**Input:** HBa3.1

**Variables:** *JCHAN, ITYPE, IDNO1, IDNO2, LHTYPE, LOHI, ZERO, SPAN1, SM1 [, CA1, BC1], SPAN2, SM2 [, CA2, CB2, SERNO, GAS, RANGE1, RANGE2] X*
**Units:** gas concentration - output units of the analyzer, typically volume %
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the analyzer channel number.

*ITYPE* is the analyzer type code. Use this input format only if ITYPE = 2, double range calibration curve.

*IDNO1* is the first curve number of analyzer and range to be used. If IDNO1 is equal to zero, analyzer is not in catalog (enter CA1, CB1, GAS, and RANGE1).

*IDNO2* is the second curve number of analyzer and range to be used. If IDNO2 is equal to zero, analyzer is not in catalog (enter CA2, CB2, and RANGE2).

*LHTYPE*: If equal to 1, range 1 and range 2 are identified by the output values of another channel: If the output of the channel is less than 1, range 1 will be used. If the output of the channel is greater than or equal to 1, range 2 will be used. If LHTYPE is equal to 2, switch from range 1 to range 2 at a particular scan. If LHTYPE is equal to 3, switch from range 1 to range 2 at or after a particular time.

*LOHI*: If LHTYPE is 1, LOHI is the channel number of the switch indicator. If LHTYPE is 2, LOHI is the scan number of the first scan after the switch. If LHTYPE is 3, LOHI is the time in seconds at or after which the switch is made (the comparison is made with the time channel identified by ITYPE = 1 from data input B3 after all reduction has been performed including by conversion coefficients C, ADD, and POWER identified on data input D3).

*ZERO* is the output for zero concentration of gas. If the "zero" spans have been identified in Part F, the character "Z" may be used in place of a number.

*SPAN1* is the output for a known concentration of gas on the first range. If the "span" scans have been identified in Part F, the character "R" may be used in place of a number.

*SM1* is the meter reading for SPAN1 concentration.

*CA1,CB1* are the calibration curve coefficients. Enter these values only if IDNO1=0.

*SPAN2* is the output for a known concentration of gas on the second range. If the "span" scans have been identified in Part F, the character "R" may be used in place of a number.

*SM2* is the meter reading for SPAN2 concentration.

*CA2,CB2* are the calibration curve coefficients. Enter these values only if IDNO2=0.

*SERNO* is the analyzer serial number, enclosed in quotation marks. Enter this value only if IDNO1=0 and IDNO2=0.

*GAS* is the type of gas analyzed, enclosed in quotation marks. Enter this value only if IDNO1=0 and IDNO2=0.

*RANGE1* is the maximum concentration of gas for 1st range of analyzer enclosed in quotation marks. Enter this value only if IDNO1=0.

*RANGE2* is the same as above but for 2nd range enclosed in quotation marks. Enter this value only if IDNO2=0.

*X* = ' X' - end-of-set mark.

**Input:** HBa4

**Variables:** *JCHAN, ITYPE [, ZERO] [, SPAN], CON X*
**Units:** gas concentration - output units of the analyzer, typically volume %
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the analyzer channel number (NPDI [3]).
*ITYPE* is the analyzer type code. Use this input format only if ITYPE = 3, change in gas concentration linearly proportional to change in analyzer output.
*ZERO* is the output for zero concentration of gas. If the "zero" spans have been identified in Part F, the character "Z" may be used in place of a number. Note: if only three arguments are entered on this input, ZERO is assumed = 0.0.
*SPAN* is the output for a known concentration of gas. If the "span" scans have been identified in Part F, the character "R" may be used in place of a number. Note: if only three or four arguments are entered with this input, SPAN is set equal to the first reading of the analyzer.
*CON* is the span gas concentration.
*X* = ' X' - end-of-set mark.

**Input:** HBa5

**Variables:** *JCHAN, ITYPE, C1, R1, C2, R2 X*
**Units:** gas concentration - output units of the analyzer, typically volume %
**Format:** EVALU8 [NPDI [2])

*JCHAN* is the analyzer channel number (NPDI [3]).
*ITYPE* is the analyzer type code. Use this input format only if ITYPE = 4, gas concentration proportional to $10^{(-R/k)}$.
*C1* is the known concentration of gas, volume % (not equal to zero).
*R1* is the output for known concentration C1. If the "zero" and/or "span" scans have been identified in Part F, the character "Z" or "R" may be used in place of a number. Whether "Z" or "R" is used depends on during which set of scans ("zero" or "span") concentration C1 was used.
*C2* is the known concentration of gas different than C1, volume % (not equal to zero).
*R2* is the output for known concentration C2. If the "zero" and/or "span" scans have been identified in Part F, the character "Z" or "R" may be used in place of a number. Whether "Z" or "R" is used depends on during which set of scans ("zero" or "span") concentration C2 was used.
*X* = ' X' - end-of-set mark.

*Example*

```
GAS%
7
32  3       Z       20.9 X
33  1  20   Z  R    52.7 X
34  1  30   Z  R    93.0 X
35  3       Z       20.9 X
36  1  0  Z  R  92.2  1.997  245.511 '100324'  'CO'  '1.0' X
47  1   6   .0000618   .0702125    70.78 X
49  1  27    .0003908    R   93.0 X
```

### 8.2 PRESSURE: Pressure Transducers

The subroutine calculates static pressure from the output of pressure transducers. A static pressure probe is two sided and senses the difference in pressure between one side and the other. A typical use for the probe is to measure the pressure difference between the inside and outside of a test chamber.

For all the static pressure calculations, the calculated values replace the raw data values in the data matrix; no new channels are created.

**Input:** HBb1

**Variables:** *NSTAT*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command PRESSURE is given.

*NSTAT* is the number of channels to be converted from raw data to static pressure. Prepare NSTAT HBb2 input(s).

**Input:** HBb2

**Variables:** *JCHAN, ZERO, CON X*
**Units:** pressure - Pa
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the pressure probe channel number (NPDI [3]).
*ZERO* is the output for zero pressure difference (ambient). If the "zero" scans have been identified in Part F, the character "Z" may be used in place of a number.
*CON* is the conversion factor from output to static pressure (pascals per unit output).
*X* = ' X' - end-of-set mark.

*Example*

```
PRESSURE
2
44  Z  13.332 X
45  Z  13.332 X
```

### 8.3 SMOKE: Optical Density

The subroutine calculates the optical density per unit length from the recorded output of a smoke meter. There are three types of relationships between the optical density and the output.

Type 1 - optical density proportional to the log of the inverse of the fraction of full transmission, output decreasing for transmission decreasing.

Type 2 - same as type 1 but output increasing for transmission decreasing.

Type 3 - optical density linearly proportional to the zero-adjusted output voltage.

Type 4 - extinction coefficient linearly proportional to the zero-adjusted output voltage.

For types 1 and 2,

$$OD = \log\frac{1}{P}$$

$$P = \frac{R-R_0}{R_1-R_0} \ .$$

Note that the equation for optical density will not allow $P$ equal to zero and, thus, $R$ may not equal $R_0$. Since, in typical practice, the meter cannot sense less than one hundredth of one percent of full transmission, that value is assumed equal to zero and for any value less than that the optical density defaults to 5.0.

For types 3 and 4,

$$S = C(R - R_1)$$

where $S$ is either the optical density (type 3) or the extinction coefficient (type 4) and $C$ is the calibration factor for the smoke meter,

$$C = \frac{\text{change in } S}{\text{change in voltage output}}$$

For all the optical density calculations, the calculated values replace the raw data values in the data matrix; no new channels are created.

**Input:** HBc1

**Variables:** *NMETER*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command SMOKE is given.

*NMETER* is the number of smoke meters. Prepare NMETER HBc2 input(s).

**Input:** HBc2

**Variables:** *JCHAN, JTYPE [{, PATH} [.ICONV [,R100 [{,R0}]]]] {[,CAL]} X*
**Units:** smoke density - $m^{-1}$
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the smoke meter channel number (NPDI [3]).

*JTYPE:* If equal to 1, calculate optical density proportional to the log of the inverse of percent of full transmission; output decreasing for transmission decreasing If equal to 2, calculate as type 1 but output increasing for transmission decreasing. If equal to 3, calculate optical density linearly proportional to zero adjusted voltage. If equal to 4, calculate extinction coefficient linearly proportional to zero adjusted voltage.

*PATH* is the effective path length of smoke meter, m. Enter this value only if JTYPE is equal to 1, 2 or 3.

*ICONV:* If ICONV is not equal to zero, convert optical density to extinction coefficient for types 1, 2 and or convert extinction coefficient to optical density for type 4.

*R100* is the output for 100% transmission. If the "span" scans have been identified in Part F, the character "R" may be used in place of a number. The default value, if omitted, is the first reading not skipped.

*R0* is the output for 0% transmission. If the "zero" scans have been identified in Part F, the character "Z" may be used in place of a number. Enter this value only if JTYPE = 1 or 2. The default value, if omitted, is 0.0.

*CAL* is the change in optical density or extinction coefficient per unit change in voltage output. Enter this value only if JTYPE = 3 or 4. The default value, if omitted, is 1.0.

*X* = ' X' - end-of-set mark.

*Example*

```
SMOKE
1
770 1 1.0   X
```

### 8.4 THERMOCOUPLE: Temperature Measurement

The subroutine calculates the temperature in degrees C (Celsius), F (Fahrenheit), K (Kelvin), or R (Rankine) from the thermocouple output voltage using an eight-segment curve fit for seven different types of thermocouples: E, S, J, T, K, B and R.

The instruments to be converted are entered in blocks, identifying only the first and last instruments in the block. A block is a contiguous set of B3 data inputs. Note that the input for this command can be simplified by grouping all the thermocouple channels together in the instrument list (Data Input B3).

**Input:** HBd1

**Variables:** NTCGRP
**Format:** open (NPDI [1])

*NTCGRP* is the number of thermocouple blocks to be converted. Prepare NTCGRP HBd2 inputs.

**Input:** HBd2

**Variables:** *JLOW, JHI, [JT [, ISCALE]] X*
**Units:** Temperature - selectable units: °C, °F, °R, or K
**Format:** EVALU8 (NPDI [2])

*JLOW* is the first thermocouple instrument number of the block. (NPDI [3]).
*JHI* is the last thermocouple instrument number of the block. (NPDI [3]).
*JT* is the thermocouple type, identified by number: 1 - Type E, 2 - Type S, 3 - Type J, 4 - Type T, 5 - Type K, 6 - Type B, 7 - Type R. The default type for missing or invalid input is 5 (Type K).
*ISCALE* is the output temperature units, identified by number: 1 - °C, 2 - °F, 3 - °K, 4 - °R. The default units for missing or invalid input is 1 (°C).
*X* = ' X' - end-of-set mark.

*Example*

```
TC
1
60   99   X
```

### 8.5 VELOCITY: Gas Velocity

The subroutine calculates the gas velocity from the output of a velocity probe (pressure transducer). There are two kinds of probes: mono-directional (pitot tube) and bi-directional. Essentially, the probes sense the dynamic air pressure on them and the transducers convert it to a voltage output.

**Pressure Calculation**

The conversion factor for changing the voltage output to pressure (the sensitivity) must be known (e.g., 10V = 1" H2O = 248.7 Pa). The pressure, *P*, is then

$$P = S(R-R_0) .$$

### Gas Density Calculation

In addition to the pressure, the gas density, $\rho$, is also necessary. The gas density is directly proportional to the absolute gas temperature: by the ideal gas law

$$\rho = \frac{\rho_0 T_0}{T} .$$

### Input Temperature

The input temperature for the gas density calculation may be from one of three different sources:

1. A single temperature from one, or the average of several, thermocouples located near the velocity probe;
2. An arbitrary, constant temperature supplied by the user;
3. An interpolated temperature from the values of two thermocouples near the velocity probe.

### Velocity Calculation

For mono-directional probes (pitot tubes), the velocity, v, is calculated by

$$v = \sqrt{2P/\rho} .$$

Maintaining appropriate units and using the gas constant of 353 kgK/m$^3$,

$$Pa = \frac{N}{m^2} = \frac{kg \times m}{s^2 \times m^2}$$

$$\rho = \frac{\rho_0 T_0 \ (K \times kg/m^3)}{T \ (K)} = \frac{353}{T}(kg/m^3) .$$

Substituting the proper units back into the velocity equation yields

$$v = 0.0753\sqrt{P\ T} \ \ (\mathrm{m/s}) .$$

The equation for bi-directional probes is similar

$$v = 0.0698\sqrt{P\ T} \ \ (\mathrm{m/s}) .$$

Note that the bi-directional probes measure velocity in two directions. In this subroutine, which direction is positive and which is negative has no real meaning. However, for the command FLOW-RATE, the convention is that the positive direction is out of the test chamber and the negative direction is into the test chamber. Mono-directional probes should be consistent with any bi-directional probes and/or with each other.

This command replaces the raw data in the data matrix with the calculated gas velocities; no new channels are created.

**Input:** HBe1

**Variables:** *NPROBE*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command VELOCITY is given.

*NPROBE* is the number of velocity probes. Prepare NPROBE HBe2 input(s).

**Input:** HBe2

**Variables:** *JCHAN, SEN, ZERO, JPTYPE, JTMETH,* {, *JTEMP*} {, *CTEMP*} {, *POSJ, JTEMP1, PTEMP1, JTEMP2, PTEMP2*} *X*
**Units:** position - m, pressure - Pa, temperature - °C, velocity - m/s
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the velocity probe channel number (NPDI [3]).

*SEN* is the sensitivity (or calibration) of transducer (Pa per output unit).

*ZERO* is the transducer output for no-flow condition. If the "zero" scans have been identified in Part F, the character "Z" may be used in place of a number.

*JPTYPE* is the velocity probe type code. If equal to 1, calculate mono-directional probe with positive change in output for positive change in gas flow. If equal to -1, calculate mono-directional probe with negative change in output for positive change in gas flow. If equal to 2, calculate mono-directional probe with positive change in output for negative change in gas flow. If equal to -2, calculate mono-directional probe with negative change in output for negative change in gas flow. If equal to 3, calculate bi-directional probe with positive change in output for positive change in gas flow. If equal to -3, calculate bi-directional probe with negative change in output for negative change in gas flow. Note: by convention, if the gas flow rate is being measured in and out of a test chamber, the out-flow is assumed to be in the positive direction. If only one direction is being investigated, assume it to be the positive direction.

*JTMETH* is the velocity probe temperature correction method code. If equal to 1, use a single temperature from a thermocouple near the velocity probe. If equal to 2, use an arbitrary, fixed temperature. If equal to 3, use an interpolation of temperature between two thermocouples near the velocity probe.

*JTEMP* is the a single temperature channel number (NPDI [3]). Enter this value only if JTMETH is equal to 1.

***CTEMP*** is the an arbitrary constant temperature. Enter this value only if JTMETH is equal to 2.

***POSJ*** is the position of velocity probe relative to an arbitrary origin. Enter this value only if JTMETH is equal to 3.

***JTEMP1*** is the upper temperature channel number (by probe position) used in interpolation (NPDI [3]). Enter this value only if JTMETH is equal to 3.

***PTEMP1*** is the position of JTEMP1 relative to same origin as probe. Enter this value only if JTMETH is equal to 3.

***JTEMP2*** is the lower temperature channel number (by probe position) used in interpolation (NPDI [3]). Enter this value only if JTMETH is equal to 3.

***PTEMP2*** is the position of JTEMP2 relative to same origin as probe. Enter this value only if JTMETH is equal to 3.

***X*** = ' X' - end-of-set mark.

*Example*

```
VELOCITY
3
51  13.332  Z  1  1  61 X
52  13.332  Z  1  1  62 X
53  13.332  Z  1  1  63 X
```

### 8.6 WT-LOSS: Total Weight Loss

The subroutine calculates the total weight loss of an item that is being weighed by a load cell. The change in weight is linearly proportional to the change in output and the total weight loss is defined as

$$m = C(R_1 - R) \ .$$

If the initial weight of the weighed item and the output, $R_0$, for zero weight are known, $C$ can be calculated by the program as

$$C = \frac{m_0}{R_1 - R_0}$$

otherwise $C$ must be input.

For all the total weight loss calculations, the calculated values replace the raw data values in the data matrix; no new channels are created.

**Input:** HBf1

**Variables:** *NCHAN*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command WT-LOSS is given.

*NCHAN* is the number of weight loss channels to be calculated. Prepare NCHAN HBf2 input(s).

**Input:** HBf2

**Variables:** *JCHAN {, ZERO, WEIGHT} {, CON} X*
**Units:** weight - kg
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the load cell channel number (NPDI [3]).
*ZERO* is the output for zero weight. If the "zero" scans have been identi- fied in Part F, the character "Z" may be used in place of a number. Enter this value only if it and the initial weight are known.
*WEIGHT* is the initial total weight of items on the load cell. Enter this value only if it and the output for zero weight are known.
*CON* is the the output to weight conversion factor. Enter this value only if ZERO or WEIGHT is unknown or CON is preferred.
*X* = ' X' - end-of-set mark.

*Example*

```
WT-LOSS Calculate total weight loss of specimen on load platform
1
16 192.6   X
```

## 9 Control Input for Complex Commands

The complex class calculation requires at least two sets of instrument values or other information, such as instrument position. Typically, the complex class calculation creates one or more new channels in order to store the calculated results.

### 9.1 BALANCE: Rate of Heat Release by an Energy Balance

The subroutine solves an energy balance equation to find the rate of heat release of the system:

$$\dot{E}_i+\dot{q}-\dot{q}_s-\dot{q}_o\dot{E}_o = 0 \; .$$

Rearranging and solving for the rate of heat release of the system we have:

$$\dot{q} = \dot{q}_s+\dot{q}_o+M_o\dot{E}_o-M_i\dot{E}_i \; .$$

The calculated results are stored in a channel created by the program. The subroutine does not reduce any raw data.

**Input:** HCa1

**Variables:** *JEIN(1), JEOUT(1), JEIN(1), JEOUT(2), . . . , JEIN(i), JEOUT(i) X*
*$1 \leq i \leq 5$*
**Units:** energy flow rate - kW
**Format:** EVALU8 (NPDI [2])

Enter convective energy flow rate in and flow rate out channel numbers in pairs.

*JEIN(i)* is the channel number(s) of the convective energy flow rate into chamber (NPDI [3]).
*JEOUT(i)* is the channel number(s) of the convective energy flow rate out of chamber (NPDI [3]).
*X* = ' X' - end-of-set mark.

**Input:** HCa2

**Variables:** *JWALL(1), JWALL(2), . . . , JWALL(i)* *$1 \; i \leq 10$*
**Units:** heat loss rate - kW
**Format:** EVALU8 (NPDI [2])

*JWALL(i)* is the channel number(s) of the total radiative rate of heat loss to surfaces (NPDI [3]).
*X* = ' X' - end-of-set mark.

**Input:** HCa3

**Variables:** *JOPEN(1), JOPEN(2), . . . , JOPEN(i)* *1 ≤ i ≤ 5*
**Units:** heat loss rate - kW
**Format:** EVALU8 (NPDI [2])

*JOPEN(i)* is the channel number(s) of the total radiative rate of heat loss through an opening (NPDI [3]).
*X* = ' X' - end-of-set mark.

*Example*

```
BALANCE
130 131 132 133 X
143 144 145 147 148 X
156 157 X
```

### 9.2 FLOW-RATE: Flow Through Openings

The subroutine makes four separate calculations based on the gas velocity profile of an opening in the test chamber: 1) neutral plane height of the opening; 2) volume flow rate in and out of the opening; 3) mass flow rate in and out of the opening; and 4) convective energy transfer rate in and out of the opening. The seven different calculated values are all stored in their own channels created by the program. This subroutine does not reduce any raw data.

#### Neutral Plane

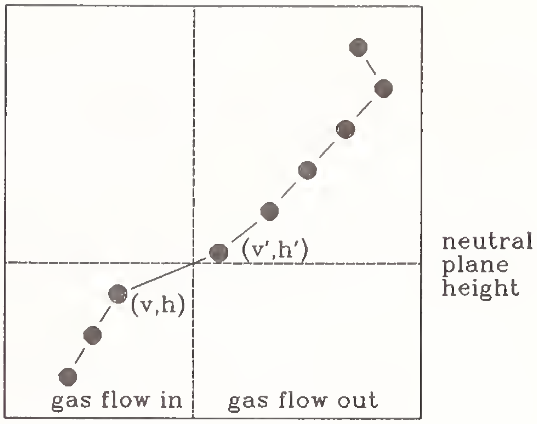
The neutral plane height calculation requires a velocity profile of the opening being investigated. The velocities are checked from top to bottom for gas flow reversal. When reversal is found, the neutral plane height is calculated by interpolating the two velocities ($v$ and $v'$) and the positions of the velocity measurements ($h$ and $h'$) to find the position of zero velocity (the neutral plane).

$$h_{np} = h + \frac{-v(h'-h)}{v'-v}$$

### Volume, Mass, and Convective Energy Flow Rates

For all three kinds of calculated flow rates, the flow rate is proportional to the product of the velocity, the opening height, and the opening width:

$$\text{volume flow rate, } V = v\ H\ W$$
$$\text{mass flow rate, } M = V\ \rho$$
$$\text{energy flow rate, } E = M\ C_p\ \Delta T$$



For air,

$$C_p = 0.9126+2.577\times10^{-4}T-3.974\times10^{-8}T^2$$

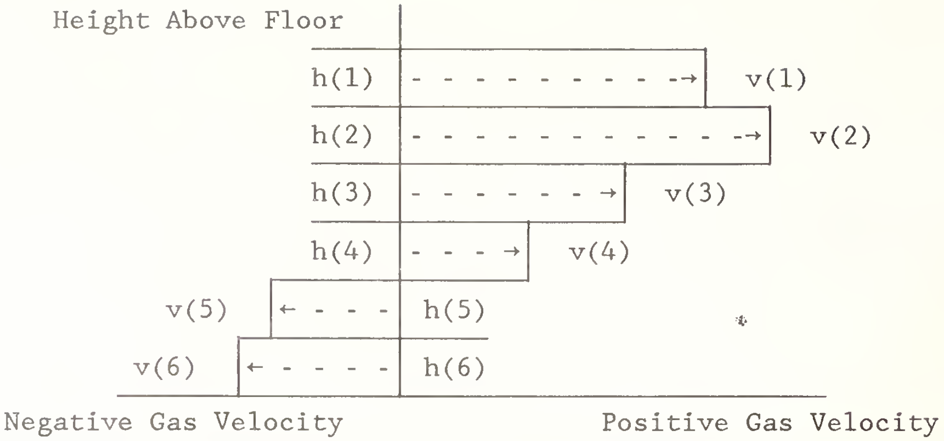
Both the mass and energy flow rate calculations require the gas temperature to make the calculations (to find p and dT respectively). The input temperature may be from three different sources:

1. an arbitrary constant temperature supplied by the user;
2. a single temperature from one, or the average of several, thermocouple(s) located near the velocity probe;
3. an interpolated temperature from the values of two thermocouples near the velocity probe.

When only one velocity in the opening is known, the calculations are made assuming the same gas velocity over the whole area. However, when a velocity profile is used, the opening height is divided into segments calculated by the program and determined by the velocity probe positions. The flow rates for each part are calculated using the opening width, the segment height, and the segment gas velocity.

There are some conventions built in to the default flow coefficient conditions of this subroutine. In particular, the default flow coefficient for the positive direction corresponds to flows of relatively hot gas at relatively high velocities while the default flow coefficient for the negative direction corresponds to flows of relatively cool gas at relatively low velocities. Thus the default conditions correspond to in = negative and out = positive. The user is not restricted to these conventions however, and may change them with the appropriate choices for the flow coefficients.

Note also that for comparison and plotting purposes, the absolute values of negative flow rates are returned. It is up to the user to decide the convention he wishes to use for in and out (-,+ or +,-). Note, however, that the negative value is stored in the first channel created and the positive value in the second.



**Input:** HCb1

**Variables:** *NPROBE, INP, IVP, IMF, IEF, HDOOR,* {, *WDOOR*} *[, FCNEG [,FCPOS]] X*
**Units:** energy flow rate - kW, height - m, mass flow rate, kg/s, volume flow rate, $m^3/s$, width - m
**Format:** EVALU8 (NPDI [2])

Only one of these inputs is read each time the command FLOW-RATE is given.

*NPROBE* is the number of velocity channels to be used by this subroutine. Prepare NPROBE HCb2 inputs.
*INP* is the neutral plane height flag. If INP is equal to zero, do not store the result; if not equal to zero, store the result (1 channel created).
*IVF* is the volume flow rate flag. If IVF is equal to zero, do not store the result(s); if equal to one, store only the flows in the negative direction (1 channel created); if equal to two, store only the flows in the positive direction (1 channel created); if equal to three, store flows in both directions (2 channels created).
*IMF* is the mass flow rate flag. The values for IMF have the same function as IVF above.
*IEF* is the convective energy flow rate flag. The values for IEF have the same function as IVF above.
*HDOOR* is the height of instrumented door or other opening.
*WDOOR* is the width of instrumented door or other opening. Enter this value only if IVF, IMF or IEF is greater than zero.
*FCNEG* is the empirically derived flow coefficient, default = 0.68.
*FCPOS* is the empirically derived flow coefficient, default = 0.73.
*X* = ‘ X’ - end-of-set mark.

**Input:** HCb2

**Variables:** ***JVEL, POSJ {, CTEMP} {,JTEMP} {, JTEMP1, POSTC1, JTEMP2, POSTC2} [, CTAMB or ITAMB] X***
**Units:** position - m, temperature - °C, velocity - m/s,
**Format:** EVALU8 (NPDI [2])

Information on all velocity probes used in above calculations.

***JVEL*** is the velocity channel number, (NPDI [3]).
***POSJ*** is the position of probe relative to bottom of opening.
***CTEMP*** is the constant temperature at the velocity probe. Enter this value only if a constant temperature correction is used.

***JTEMP*** is thermocouple channel number. Enter this value only if the temperature correction is from a single, related thermocouple (NPDI [3]).
***JTEMP1*** is the upper temperature channel number (by probe position) used in interpolation, (NPDI [3]). Enter this value only if using temperature interpolation.

***POSTC1*** is the position of JTEMP1 relative to same origin as probe. Enter this value only if using temperature interpolation.
***JTEMP2*** is the lower temperature channel number (by probe position) used in interpolation, (NPDI [3]). Enter this value only if using temperature interpolation.

***POSTC2*** is the position of JTEMP2 relative to same origin as probe. Enter this value only if using temperature interpolation.
***CTAMB*** is the ambient temperature.

***ITAMB*** is any integer value. The ambient temperature will be the first reading from whichever temperature method is used above. Note that if neither CTAMB nor ITAMB are entered, the current default value for the ambient temperature will be used (20 °C, unless redefined).
***X*** = ' X' - end-of-set mark.

*Example*

```
FLOW-RATE
6 1 0 3 0 2.03 0.8 0.68 0.73 X
39 0.05 60 1 X
40 0.4  61 1 X
41 0.8  62 1 X
42 1.2  63 1 X
43 1.6  64 1 X
44 2.0  65 1 X
```

### 9.3 GAS-FLOW: Mass Flow Rate of a Gas

The subroutine calculates the mass flow rate of a gas of known concentration, molecular weight, and velocity through an opening of known area. There are two methods of calculating the mass flow rate: one is for openings where only one gas concentration value is available and the other is where there is more than one value (vertical profile). Both use the same equation to calculate the mass flow rate:

$$\dot{m} = \frac{x_g W_g v A T_{amb} \rho_{amb}}{W_a T_g} \ .$$

When more than one gas probe is used (method 2), the opening is divided into segments, the sizes of which are governed by the positions of the probes. The boundary between two probes is simply the halfway point. The total mass flow rate of the gas is the sum of the mass flow rates for the segments. Note that there is provision for the user to select segment sizes if desired.

The calculated values are stored in channels created by the program. This subroutine does not reduce any raw data.

**Input:** HCc1

**Variables:** *ICALC, [CTAMB or JTAMB] IGAS X*
**Units:** gas concentration - volume percent, height - m, mass flow rate - kg/s, temperature - °C
**Format:** EVALU8 (NPDI [2])

Only one set of these inputs is read each time the command GAS-FLOW is given. One channel is created for every HCc1 input read.

*ICALC* is the calculation type code. If ICALC is equal to 1, use a single segment opening. Prepare one HCc2 input. IF ICALC is equal to 2, use a multi-segment opening. Prepare one HCc3 input, and one set of HCc4 inputs.
*CTAMB* is the constant ambient temperature.
*JTAMB* is the channel number of ambient temperature (NPDI [3]).
Note that if both CTAMB and JTAMB are left blank, the current default value for ambient temperature will be used (20 °C, unless redefined).
*IGAS* is a 20 character name for the kind of gas to be quantified.
*X* = ‘ X’ - end-of-set mark.

**Input:** HCc2

**Variables:** *JGAS, JVEL1, CTEMP or JTEMP1, AREA, AMWG X*
**Units:** area - m$^2$, gas concentration - volume percent, molecular weight - kg/kmol, temperature - °C, velocity - m/s
**Format:** EVALU8 (NPDI [2])

Read this input only if ICALC is equal to 1.

*JGAS* is the gas concentration channel number (NPDI [3]).
*JVEL1* is the gas velocity channel number (NPDI [3]).
*CTEMP* is a constant associated temperature.
*JTEMP1* is the gas temperature channel number (NPDI [3]).
*AREA* is the cross sectional area of opening.
*AMWG* is the molecular weight of gas.
*X* = ' X' - end-of-set mark.

**Input:** HCc3

**Variables:** *NGAS, HT, WD, JNP, ISMETH, AMWG X*
**Units:** area - m$^2$, height - m, molecular weight - kg/kmol, width - m
**Format:** EVALU8 (NPDI [2])

Read this input only if ICALC is equal to 2.

*NGAS* is the number of gas probe locations. Prepare NGAS HCc4 inputs.
*HT* is the height of opening parallel to line of probes.
*WD* is the width of opening perpendicular to line of probes.
*JNP* is the neutral plane height channel number (NPDI [3]).
If *ISMETH* is equal to 1, the subroutine determines fixed segment heights based on gas probe positions (entered with input HCc4) If ISMETH is equal to 2, the user supplies fixed segment heights (one for each gas probe). See SEGHT (input HCc4).
*AMWG* is the molecular weight of gas.
*X* = ' X' - end-of-set mark.

**Input:** HCc4

**Variables:** *JGAS, GASPOS, IVMETH, ITMETH {,JVEL} {,JVEL1, VPOS1, JVEL2, VPOS2} {,CTEMP or JTEMP} {,JTEMP1, TPOS1, JTEMP2, TPOS2} {,SEGHT} X*

**Units:** area - $m^2$, gas concentration - volume percent, height - m, temperature - °C, velocity - m/s, width - m

**Format:** EVALU8 (NPDI [2])

Read this input only if ICALC is equal to 2.

*JGAS* is the gas concentration channel number (NPDI [3]).

*GASPOS* is the position of gas probe from bottom of opening.

If *IVMETH* is equal to 1, use single associated velocity channel number. If IVMETH is equal to 2, use velocity interpolation.

If *ITMETH* is equal to 1, use single associated temperature channel number or constant temperature IF ITMETH is equal to 2, use temperature channel interpolation.

*JVEL* is a single associated velocity channel number (NPDI [3]). Enter this value only if IVMETH is equal to 1.

*JVEL1* is the upper velocity channel number (by probe position) used in interpolation (NPDI [3]). Enter this value only if IVMETH is equal to 2.

*VPOS1* is the position of JVEL1 from bottom of opening. Enter this value only if IVMETH is equal to 2.

*JVEL2* is the lower velocity channel number used in interpolation (NPDI [3]). Enter this value only if IVMETH is equal to 2.

*VPOS2* is the position of JVEL2 from bottom of opening. Enter this value only if IVMETH is equal to 2.

*CTEMP* is a constant associated temperature. Enter this value only if ITMETH is equal to 1.

*JTEMP* is a single associated temperature channel number (NPDI [3]). Enter this value only if ITMETH is equal to 1.

*JTEMP1* is the upper temperature channel number (by probe position) used in interpolation (NPDI [3]). Enter this value only if ITMETH is equal to 2.

*TPOS1* is the position of JTEMP1 from bottom of opening. Enter this value only if ITMETH is equal to 2.

*JTEMP2* is the lower temperature channel number used in interpolation (NPDI [3]). Enter this value only if ITMETH is equal to 2.

*TPOS2* is the position of JTEMP2 from bottom of opening. Enter this value only if ITMETH is equal to 2.

*SEGHT* is the user supplied height of segment associated with JGAS. Enter this value only if ISMETH is greater than 1 (see input HCc3).

*X* = ‘ X’ - end-of-set mark.

*Example*

```
GAS-FLOW  MASS FLOW RATE OF CO THROUGH STACK
1 'CO' X
781 $06 $05 1.486 28. X
```

### 9.4 HEAT-RATE: Heat Release Rate by Oxygen Consumption

The large-scale measurement which has benefitted the most from the emergence of science in large-scale fire testing is the measurement of the rate of heat released by a fire. With few exceptions [3],[4], this is calculated by the use of the oxygen consumption principle. If all the exhaust from a room fire test is collected, measurement of temperature, velocity, and oxygen, carbon dioxide, carbon monoxide, and water vapor concentrations in the exhaust collection hood can be used to estimate the rate of energy production of the fire. With these measurements, the total rate of heat release from the room can be determined from [5]:

$$\dot{q} = \left( E\phi - \left(E_{CO}-E\right)\frac{1-\phi}{2}\frac{X_{CO}}{X_{O_2}} \right)\frac{M_{O_2}}{M_a}\ \dot{m}_a \left(1-X_{H_2O}\right) X^o_{O_2}$$

where

$$M_e = \left(1-X_{H_2O}\right)\left(X_{O_2}+4X_{CO_2}+2.5\right)\ 4+18$$

$$\dot{m}_e = C\sqrt{\frac{M_{dry}\Delta p}{M_e T_e}}$$

$$\frac{\dot{m}_a}{M_a} = \frac{\dot{m}_e}{M_e}\frac{\left(1-X_{H_2O}\right)\left(1-X_{O_2}-X_{CO_2}-X_{CO}\right)}{\left(1-X^o_{H_2O}\right)\left(1-X^o_{O_2}-X^o_{CO_2}\right)}$$

$$\phi = \frac{X^o_{O_2}\left(1-X_{CO_2}-X_{CO}\right)-X_{O_2}\left(1-X^o_{CO_2}\right)}{X^o_{O_2}\left(1-X_{O_2}-X_{CO_2}-X_{CO}\right)}\ .$$

Simplifications are available, with some loss of precision, if concentrations of some of the gas species are not measured [6].

Figure 3 shows an example of calculated heat release rate from several large scale fire tests [9]. Measurement errors in rate of heat release measurements can be higher than in other measurements, especially for smaller fires. In one study [7], coefficients of variation ranged from 4 to 52 percent. With an oxygen depletion for a 100 kW fire of only 0.26 percent, the calculation of heat release rate suffers the same fate as the calculation of mass flows with pressure probes described above, with much of the uncertainty in the heat release calculations attributable to noise in the underlying measurements.

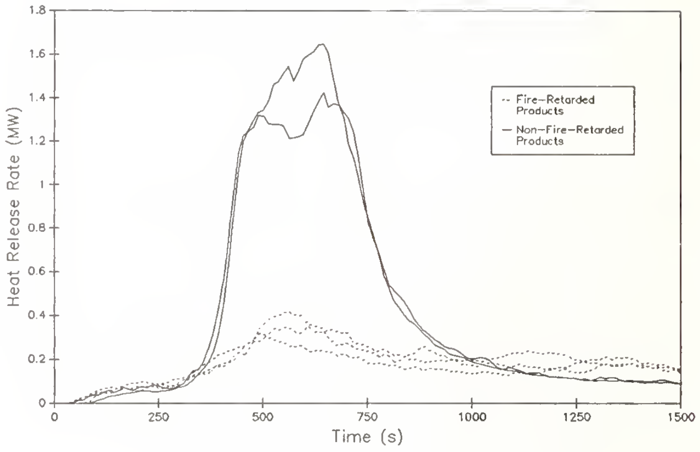


Figure 3. An example of heat release rate calculated from oxygen consumption calorimetry in several large scale fire tests[9].

This technique has been used extensively in both small and large-scale testing [8], [9], [10], [11], [12]. Babrauskas [10], for instance, has demonstrated the validity of the measurements in a study of upholstered furniture fires. He provides comparisons between replicate tests in the open and enclosed in a room. He notes precision to within 15 percent for fires of 2.5 MW and consistent comparisons of heat release rate expected from mass loss measurements to those measured by oxygen consumption calorimetry.

This command calculates the total rate of heat release for a system based on the oxygen depletion of the gas leaving the system [5] as described above. Some or all of the following input variables are required to make the calculations: oxygen, carbon dioxide, carbon monoxide and water vapor concentrations (volume percent), exhaust gas velocity, exhaust opening (doorway, window, stack, etc.) dimensions, exhaust gas temperature, and input air volume flow rate.

The calculated values are stored in their own channels created by the program. One channel is created each time the command HEAT-RATE is given. This subroutine does not reduce any raw data.

**Input:** HCe1

**Variables:** *JO2, JMFR [[[,ICO2 {,JCO2}] [,ICO {,JCO}]] [IH2O {,JH2O}]] [[,ALPHA], E] X*
**Units:** combustion expansion factor - dimensionless, gas concentration - volume percent, heat released per kg of oxygen consumed - kJ/kg, heat release rate - kW, mass flow rate - kg/s
**Format:** EVALU8 (NPDI [2])

*JO2* is the oxygen concentration channel number (NPDI [3]).
*JMFR* is a mass flow rate channel number (NPDI [3]).
If *ICO2* is equal to zero, no carbon dioxide correction is made. If ICO2 is greater than zero, oxygen is corrected for carbon dioxide.
*JCO2* is the carbon dioxide concentration channel number used to correct the oxygen concentration (NPDI [3]). Enter this value only if ICO2 is greater than zero.
If *ICO* is equal to zero, no carbon monoxide correction is made. If ICO is greater than zero, oxygen is corrected for carbon monoxide.
*JCO* is the carbon monoxide concentration channel number used to correct the oxygen concentration (NPDI [3]). Enter this value only if ICO is greater than zero.
If *IH2O* is equal to zero, no water vapor correction is made. If IH2O is greater than zero, oxygen is corrected for water vapor.
*JH2O* is the carbon dioxide concentration channel number used to correct the oxygen concentration (NPDI [3]). Enter this value only if IH2O is greater than zero.
*ALPHA* is the combustion expansion factor, dimensionless. (Default = 1.105.)
*E* is the net heat release by complete combustion per kg of oxygen consumed. (Default = 13100 kJ/kg.)
*X* = ‘ X’ - end-of-set mark.

*Example*

```
HEAT-RATE
49 58 1 50 0 0 1.6 12400. X
```

### 9.5 HOT/COLD: Two-Zone Layer Interface and Temperature

The subroutine finds the height of the hot/cold interface and the average temperatures above and below the interface using a temperature profile of the gas [13].

The temperature at the interface is defined to be:

$$T_i = T_l+(T_{he}-T_l)C$$

Once the temperature at the interface height is known, the height of the interface is found by interpolating between the pair of profile temperatures that bracket the interface temperature. Note that you may request the program to calculate (by extrapolation) the temperature at height equal to

0.0 and that the extrapolated temperature and position are then available to the algorithm for finding the interface height.

Once the interface height is known, the average temperatures above and below it are calculated.

The calculated interface height, average upper gas temperature, and average lower gas temperature are stored in channels created by the program, at the option of the user. This subroutine does not reduce any raw data. Any units may be used for temperature and position as long as they are consistent among themselves. Only one set of HCf inputs is read each time the command HOT/COLD is given.

**Input:** HCf1

**Variables:** *IHT, IHOT, ICOLD, ROOMHT [,IXTRAP,] [PCT] X*
**Units:** height - m, temperature - °C
**Format:** EVALU8 (NPDI [2])

If *IHT* is greater than zero, store the calculated interface height. One channel is created.
If *IHOT* is greater than zero, store the calculated temperature of the gases above the interface.
If *ICOLD* is greater than zero, store the calculated temperature of the gases below the interface.
*ROOMHT* is the the height of the ceiling at the point in line with the temperature profile.
If *IXTRAP* is greater than zero, calculate temperature at height equals 0.0 by extrapolation of the bottom two thermocouples.
*PCT* is the temperature at the interface height, defined as $T_l + (T_{he} - T_l) * (PCT/100)$. The default value is 15.
*X* = ‘ X’ - end-of-set mark.

**Input:** ICf2

**Variables:** *JTEMP(1), JTEMP(2), . . . , JTEMP(i) X* $1 \le i \le 25$
**Format:** EVALU8 (NPDI [2])

*JTEMP(i)* is the thermocouple channel number (NPDI [3]). Thermocouples may be in any order. The positions, entered with data input HCf3 below, must be in the same order.
*X* = ‘ X’ - end-of-set mark.

**Input:** HCf3

**Variables:** *POS(1), POS(2), . . . , POS(i) X* $1 \leq i \leq 25$
**Units:** height - m
**Format:** EVALU8 (NPDI [2])

*POS(i)* is the height of the thermocouple in the corresponding position on data input HCf2.
*X* = ‘ X’ - end-of-set mark.

*Example*

```
HOT/COLD  NORTHWEST CORNER OF BURN ROOM (STRING #1)
1 1 1 2.44 X
341    342    343    344    346    347    348    349  X
0.15   0.60   0.91   1.23   1.83   2.13   2.30   2.44 X
```

### 9.6 MASS-FLOW: Mass Flow and Neutral Plane Height From Temperature Measurements

The subroutine calculates the height of the neutral plane from the temperature profile at the room opening and one static pressure probe and the mass flow rate of hot gas in and out of the room of interest based only on the gas temperature and opening dimensions [14, 15, 16, 17].

The neutral plane is found by solving the following equation for “$H_{np}$”:

$$\Delta P + g\rho \int_{H_p}^{H_{np}} 1 - \frac{T_{amb}}{T}\, dH = 0 \; .$$

Once the neutral plane height is calculated, the mass flow rates are calculated using the following equation:

$$\dot{m}_{inflow} = CW\rho T_{amb}\sqrt{2g} \int_{H_{np}}^{0} \sqrt{\frac{1}{T_d}\int_{H_{np}}^{H}\left(\frac{1}{T_o}-\frac{1}{T_i}\right)dh_i}\quad dh_d$$

$$\dot{m}_{outflow} = CW\rho T_{amb}\sqrt{2g} \int_{H_{np}}^{H_{door}} \sqrt{\frac{1}{T_d}\int_{H_{np}}^{H}\left(\frac{1}{T_o}-\frac{1}{T_i}\right)dh_i}\quad dh_d \; .$$

The results are stored in channels created by the program. This subroutine does not reduce any raw data.

**Input:** HCg1

**Variables:** *WDOOR, HDOOR, HINT, INP, IMFI, IFMO, CPR or JPR, PPR, [CTAMB, or JTAMB [,FCI [,FCO [,HACC]]]] X*
**Units:** height - m, mass flow rate - kg/s, pressure - Pa, temperature - °C, width - m
**Format:** EVALU8 (NPDI [2])

*WDOOR* is the width of opening.
*HDOOR* is the height of opening.
*HINT* is the height of interior.
If *INP* is not equal to 0, save the neutral plane heights calculated. One channel is created.
If *IMFI* is not equal to 0, calculate and save the mass flow rate into room. One channel is created.
If *IMFO* is not equal to 0, calculate and save the mass flow rate out of room. One channel is created.
*CPR* is the constant static pressure.
*JPR* is the static pressure channel number (NPDI [3]).
*PPR* is the position of static pressure measurement above floor.
*CTAMB* is the constant ambient temperature.
If *JTAMB* is less than 0, use default ambient temperature (20 °C, unless redefined). If JTAMB is greater than or equal to 0, use a channel of temperature readings as the ambient (NPDI [3]).
*FCI* is the flow coefficient for flows going into room (default = 0.68).
*FCO* is the flow coefficient for flows going out of room (default = 0.73).
*HACC* is the accuracy of the neutral plane height calculation (default = 0.01 m).
*X* = ‘ X’ - end-of-set mark.

**Input:** HCg2

**Variables:** *JTCI(1), JTCI(2), . . . , JTCI(i) X* *$2 \leq i \leq 50$*
**Units:** temperature - °C
**Format:** EVALU8 (NPDI [2])

*JTCI* is the interior thermocouple channel number (NPDI [3]).
*X* = ‘ X’ - end-of-set mark.

**Input:** HCg3

**Variables:** *PI(1), PI(2), . . . , PI(i) X* $2 \le i \le 50$  
**Units:** height - m  
**Format:** EVALU8 (NPDI [2])

*PI(i)* is the interior thermocouple positions (height above floor) of above thermocouples in same order.  
*X* = ‘ X’ - end-of-set mark.

**Input:** HCg4

**Variables:** *JTCD(1), JTCD(2), . . . , JTCD(i) X* $2 \le i \le 50$  
**Units:** temperature - °C  
**Format:** EVALU8 (NPDI [2])

*JTCD(i)* is the doorway thermocouple channel number (NPDI [3]).  
*X* = ‘ X’ - end-of-set mark.

**Input:** HCg5

**Variables:** *PD(1), PD(2), . . . , PD(i) X* $2 \le i \le 50$  
**Units:** height - m  
**Format:** EVALU8 (NPDI [2])

*PD(i)* is the doorway thermocouple positions (height above floor) of above thermocouples in same order.  
*X* = ‘ X’ - end-of-set mark.

**Input:** HCg6

**Variables:** *CTCO or JTCO(1), JTCO(2), . . . , JTCO(i) X* $1 \le i \le 50$  
**Units:** temperature - °C  
**Format:** EVALU8 (NPDI [2])

*CTCO* is the connected room (or outside) constant temperature. Do NOT prepare image HCg7.  
If *JTCO* is less than 0, use default ambient temperature. Do NOT prepare image HCg7. If JTCO is greater than 0, use connected room thermocouple channel number (NPDI [3]). If i = 1, do NOT prepare image HCg7.  
*X* = ‘ X’ - end-of-set mark.

**Input:** HCg7

**Variables:** *PO(1), PO(2), . . . , PO(i) X* *2 ≤ i ≤ 50*
**Units:** height - m
**Format:** EVALU8 (NPDI [2]}

*PO(i)* is the connected room (or outside) thermocouple positions (height above floor) of above thermocouples in same order.
*X* = ' X' - end-of-set mark.

*Example*

```
MASS-FLOW  AT BURN ROOM DOOR
0.76 2.03 2.44 1 1 1 399 0.03 X
341    342    343    344    345    346    347    348    349   X
0.15   0.60   0.91   1.23   1.52   1.83   2.13   2.30   2.44 X
300    301    302    303    304    305    306    307    308    309   X
0.15   0.30   0.61   0.76   0.91   1.07   1.22   1.37   1.52   1.83 X
340    321    322    323    324    325    326    327    328    329    334   X
0.15   0.66   0.97   1.11   1.26   1.42   1.57   1.89   2.03   2.15   2.44 X
```

### 9.7 MASS-FLOW-2: Mass Flow From Temperature Measurements

The subroutine is very similar to the one used for the command MASS-FLOW, except that the neutral plane height is an input value instead of a calculated value. The mass flow rate of gas in and out of the fire room is still based on the gas temperature and the opening dimensions. The results are stored in channels created by the program. This subroutine does not reduce any raw data.

**Input:** HCh1

**Variables:** *WDOOR, HDOOR, HINT, CNP or JNP, IMFI, IFMO, CTAMB or JTAMB [,FCI [,FCO [,HACC]]] X*
**Units:** height - m, mass flow rate - kg/s, pressure - Pa, temperature - °C, width - m
**Format:** EVALU8 (NPDI [2])

*WDOOR* is the width of opening.
*HDOOR* is the height of opening.
*HINT* is the height of interior.
*CNP* is a constant neutral plane height.
*JNP* is the neutral plane height channel (NPDI [3]).
If *IMFI* is not equal to 0, calculate and save the mass flow rate into room. One channel is created.
If *IMFO* is not equal to 0, calculate and save the mass flow rate out of room. One channel is created.
*CTAMB* is the constant ambient temperature.
If *JTAMB* is less than 0, use default ambient temperature (20 °C, unless redefined). If JTAMB is greater than or equal to 0, use a channel of temperature readings as the ambient (NPDI [3]).
*FCI* is the flow coefficient for flows going into room (default = 0.68).
*FCO* is the flow coefficient for flows going out of room (default = 0.73).
*HACC* is the accuracy of the numerical integration over the height of the opening (default = 0.01 m).
*X* = ' X' - end-of-set mark.

**Input:** HCh2

**Variables:** *JTCD(1), JTCD(2), . . . , JTCD(i) X* $2 \le i \le 50$
**Units:** temperature - °C
**Format:** EVALU8 (NPDI [2])

*JTCD(i)* is the doorway thermocouple channel number (NPDI [3]).
*X* = ' X' - end-of-set mark.

**Input:** HCh3

**Variables:** *PD(1), PD(2), . . . , PD(i) X* $2 \le i \le 50$
**Units:** height - m
**Format:** EVALU8 (NPDI [2])

*PD(i)* is the doorway thermocouple positions (height above floor) of above thermocouples in same order.
*X* = ' X' - end-of-set mark.

**Input:** HCh4

**Variables:** *JTCI(1), JTCI(2), . . . , JTCI(i) X* $2 \leq i \leq 50$
**Units:** temperature - °C
**Format:** EVALU8 (NPDI [2])

*JTCI* is the interior thermocouple channel number (NPDI [3]).
*X* = ' X' - end-of-set mark.

**Input:** HCh5

**Variables:** *PI(1), PI(2), . . . , PI(i) X* $2 \leq i \leq 50$
**Units:** height - m
**Format:** EVALU8 (NPDI [2])

*PI(i)* is the interior thermocouple positions (height above floor) of above thermocouples in same order, m.
*X* = ' X' - end-of-set mark.

*Example*

```
MASS-FLOW-2
0.76 2.03 2.44 67 1 1 399 0.03 X
300    301    302    303    304    305    306    307    308    309   X
0.15   0.30   0.61   0.76   0.91   1.07   1.22   1.37   1.52   1.83  X
341    342    343    344    345    346    347    348    349   X
0.15   0.60   0.91   1.23   1.52   1.83   2.13   2.30   2.44  X
```

### 9.8 MASS-FLOW-3: Mass Flow in a Duct

The subroutine calculates the mass flow rate of gas through an opening where the gas flow is in only one direction and the velocity, temperature, and area perpendicular to the flow are known. It is particularly well suited for calculations in an exhaust duct.

The equation for the mass flow rate is

$$\dot{m} = CvA\rho\frac{T_{amb}}{T}\ .$$

The calculated results are stored in a channel created by the program. One channel is created each time the command MASS-FLOW-3 is given. The subroutine does not reduce any raw data.

**Input:** HCi1

**Variables:** *JVEL, AREA, JTEMP or CTEMP [,C [,CTAMB or JTAMB]] X*
**Units:** area - m$^2$, temperature - °C, velocity - m/s
**Format:** EVALU8 (NPDI [2])

*JVEL* is the gas velocity channel number, (NPDI [3]).
*AREA* is the cross sectional area of the exhaust duct perpendicular to the gas flow.
*JTEMP* is the gas temperature channel number (NPDI [3]).
*CTEMP* is an arbitrary constant gas temperature.
*C* is an empirical flow coefficient. (Default is 1.0.)
*CTAMB* is an arbitrary constant ambient temperature.
*JTAMB* is an ambient temperature channel number (NPDI [3]).

Note that if both CTAMB and JTAMB are left blank, the current default value for the ambient temperature will be used (20 °C, unless redefined).

*X* = ‘ X’ - end-of-set mark.

*Example*

```
MASS-FLOW-3
245 2.489 312 0.73 20. X
```

### 9.9 STATIC: Analysis of Static Pressure Measurements

The subroutine uses static pressure measurements in conjunction with some other test parameters, such as temperature, to calculate various quantities of interest.

Generally, if more than one probe is used they are arranged such that a pressure “profile” of the room from top to bottom can be ascertained. In addition, the profile makes it possible to calculate gas velocities through openings in the chamber, the neutral plane and thermal discontinuity heights in the openings, and the interior gas temperature in the chamber.

The standard unit for pressure is the pascal. The calculated velocities, heights, and temperatures are in meters per second, meters, and degrees Celsius, respectively.

### Gas Velocities

The gas velocity, through an opening at a given height, *z*, can be calculated from the static pressure at height z and the average local gas temperature:

$$v_z = \sqrt{\frac{2\Delta P_z}{\rho_z}}$$

Using the gas law:

$$\rho_z = \rho_0 \frac{T_0}{T_z}$$

The velocity equation then becomes

$$v_z = \sqrt{\frac{2T_z\Delta p_z}{\rho_0 T_0}}$$
$$= .07526\sqrt{T_z\Delta P_z}\ .$$

### Neutral Plane and Thermal Discontinuity Heights

To find the neutral plane and thermal discontinuity heights from the static pressure "profile," two least square straight lines are found: one for the probes above the neutral plane (positive pressure difference with ambient), one for the probes below the neutral plane (negative pressure difference with ambient). The Y-axis intercept for the former yields the neutral plane height; the intersection of the two yields the thermal discontinuity height. In addition, once the thermal discontinuity height is found, the first equation can be rearranged and evaluated to find the static pressure at that height.

The equations for the two lines are of the form:

$$z = a\Delta P + b$$
$$z' = c\Delta P + d$$

where $z$ and $z'$ are the heights above the floor and dP and dP' are the pressure differences with respect to ambient static pressure at those heights.

The equations for the coefficients are:

$$a = \frac{n\sum_{i=1}^{n} \Delta P_i z_i - \sum_{i=1}^{n} \Delta P_i \sum_{i=1}^{n} z_i}{n\sum_{i=1}^{n} \Delta P_i^2 - \left(\sum_{i=1}^{n} \Delta P_i\right)^2}, \qquad b = \frac{\sum_{i=1}^{n} z_i - a\sum_{i=1}^{n} \Delta P_i}{n}$$

$$c = \frac{n'\sum_{i=1}^{n'} \Delta P_i' z_i' - \sum_{i=1}^{n'} \Delta P_i' \sum_{i=1}^{n'} z_i'}{n'\sum_{i=1}^{n'} \Delta P_i'^2 - \left(\sum_{i=1}^{n'} \Delta P_i'\right)^2}, \qquad d = \frac{\sum_{i=1}^{n'} z_i' - a\sum_{i=1}^{n'} \Delta P_i'}{n'}$$

where $n$ and $n'$ are the number of probes above and below the neutral plane, respectively. The neutral plane height, $z_{np}$, equals b. The thermal discontinuity height occurs at the height where $\Delta P = \Delta P' = dP_{td}$ and $z = z' = z_{td}$. Then, solving the two equations, $z_{td} = a * dP_{td} + b$, and $z_{td} = c * dP_{td} + d$ simultaneously for the thermal discontinuity height yields

$$z_{td} = \frac{bc-ad}{c-a} .$$

Rearranging the first equation and evaluating at height $z_{td}$ yields the pressure at the thermal discontinuity height:

$$z_{td} = a\Delta P_{td} + b$$
$$\Delta P_{td} = \frac{bz_{td}}{a} .$$

### Interior Gas Temperatures

The interior gas temperature $T$ at a given height $z$ is calculated from the difference in pressure between height $z$ and the thermal discontinuity height and the difference in position between the two heights.

$$T_z = \frac{T_{amb}}{1 - \frac{\Delta P}{g\rho_{amb}\Delta z}} - 273.15$$

where $\Delta P = dP_z - dP_{td}$ and $\Delta z = z - z_{td}$.

The calculated values are stored in channels created by the program. This subroutine does not reduce any raw data.

**Input:** HCj1

**Variables:** *NPROBE, IVEL [,JTAVG], INPTD [,HEIGHT], ITEMP [,CTAMB or JTAMB] X*
**Units:** height - m, pressure - Pa, temperature - °C, velocity m/s
**Format:** EVALU8 (NPDI [2])

*NPROBE* is the number of static pressure probes in this profile. Prepare NPROBE HCj2 inputs.

If *IVEL* is equal to zero, no calculations made. If IVEL is greater than zero, calculate air velocity through opening from interior static pressure for each probe. NPROBE channels will be created.

*JTAVG* is the average interior upper air temperature channel (NPDI [3]). Enter this value only if IVEL is greater than zero.

If *INPTD* is equal to zero, no calculations made. If INPTD is greater than zero, calculate the positions of the neutral plane and thermal discontinuity and the static pressure at the thermal discontinuity height. Three channels will be created.

*HEIGHT* is the opening height. Enter this value only if INPTD is greater than zero.

If *ITEMP* is equal to zero, no calculations made. If ITEMP is greater than, calculate the interior gas temperature near each probe. NPROBE channels will be created. ITEMP may only be greater than zero if INPTD is greater than zero.

*CTAMB* is a constant ambient temperature.

*JTAMB* is the channel number of the ambient temperature (NPDI [3]).

Note that if both CTAMB and JTAMB are left blank, the current default value for the ambient temperature will be used (20 °C, unless redefined). The ambient temperature is not required if IVEL is equal to zero and ITEMP is equal to zero.

*X* = ' X' - end-of-set mark.

**Input:** HCj2

**Variables:** *JCHAN, POS {,JTOPEN} X*
**Units:** height - m, pressure - Pa, temperature - °C
**Format:** EVALU8 (NPDI [2])

*JCHAN* is the static pressure channel number (NPDI [3]).

*POS* is the position of probe relative to floor.

*JTOPEN* is the corresponding opening thermocouple channel (NPDI [3]). Enter this value only if IVEL is greater than zero (see input HCj1).

*X* = ' X' - end-of-set mark.

*Example*

```
STATIC
6 0 1 2.03 0 46 X
40 0.05 X
41 0.4  X
42 0.8  X
43 1.2  X
44 1.6  X
45 2.0  X
```

### 9.10 SURFACE: Heat Loss and Incident Heat Flux to Surfaces

The subroutine calculates the average and total heat loss rate and the incident heat flux to a semi-infinite surface (such as a wall or ceiling) from the surface temperature and material characteristics [18]. The three equations are related to each other and are as follows:

$$\begin{aligned} \dot{q}_{a,j} &= \frac{\kappa}{2}\sqrt{\frac{\pi}{\alpha}}\frac{T-T_0}{\sqrt{t_j-t_{j-1}}}-\sum_{i=1}^{j-1}\dot{q}_{a,i}\,\sqrt{t_j-t_{i-1}}-\sqrt{T_j-t_i} \\ q_{t,j} &= \dot{q}_{a,j}A \\ q_j &= q_{t,j}+\epsilon\sigma T^4\ . \end{aligned}$$

All three calculations are made at the same time and each is stored in its own channel created by the program. This subroutine does not reduce any raw data.

**Input:** HCk1

**Variables:** *IWTYPE, AREA, CTEMP or JTEMP, JTIME {,TCONA, TCONB, ALPHA, EPSIS}, IAHLR, ITHLR, ITNHF [,CTAMB or JTAMB] X*

**Units:** area - m$^2$, heat flux - kW/m$^2$, heat loss rate - kW, temperature - °C, thermal conductivity - kW/m·K, thermal diffusivity - m$^2$/s, time - s, total heat loss - kJ

**Format:** EVALU8 (NPDI [2])

*IWTYPE* is the wall material type code. 1 = plywood, 2 = concrete block, 3 = gypsum board (default), 4 = acoustic tile, 5 = ceramic board, 6 = kaowool, 0 = any user defined material (thermal characteristics entered below).

*AREA* is the total surface area.

*CTEMP* is a constant surface temperature.

*JTEMP* is the temperature channel number (single or average) (NPDI [3]).

*JTIME* is the reference time channel number.

*TCONA,TCONB* define the thermal conductivity of a material. TCONA and TCONB are variables in the straight line definition of the thermal conductivity = TCONA + TCONB * T, where T is the material temperature. For temperatures less than 260 °C, the conductivity is assumed the same as at 260 °C. Enter these values only if IWTYPE = 0 .

*ALPHA* is the thermal diffusivity of surface material not in list, m$^2$/s. Enter this value only if IWTYPE = 0.

*EPSIS* is the total emissivity of surface material not in list. Enter this value only if IWTYPE = 0.

If *IAHLR* is greater than zero, save the calculated values of the average rate of heat loss. One channel created.

If *ITHLR* is greater than zero, save the calculated values of the total heat loss. One channel created.

If *ITNHF* is greater than zero, save the calculated values of the incident heat flux. One channel created.

*CTAMB* is a constant ambient temperature.

If *JTAMB* is equal to zero, use CTEMP or the first reading from channel JTEMP, whichever is used. If JTAMB is greater than zero, the ambient temperature channel number (NPDI [3]).

Note that if neither CTAMB nor JTAMB are used, the default ambient temperature is used. (See AMBIENTS command.)

*X* = ‘ X’ - end-of-set mark.

*Example*

```
SURFACE
3 12.19 126 0 1 1 1 X
```

### 9.11 VENT-LOSS: Radiative Heat Loss Through an Opening

The subroutine calculates the radiative heat loss rate through an opening based on the temperature in the opening. The equation is as follows:

$$\dot{q} = \sigma \epsilon A (T^4 - T_{amb}{}^4) \ .$$

If a temperature profile is used (i.e., more than one thermocouple in the opening), the opening is divided into segments. The heat loss rate for each segment is calculated and added together to get the total.

The calculated results are stored in a channel created by the program. This subroutine does not reduce any raw data.

**Input:** HCl1

**Variables:** *OPENWD, OPENHT [,CTAMB or JTAMB] X*
**Units:** height - m, temperature - °C, width - m
**Format:** EVALU8 (NPDI [2])

One channel is created each time the command VENT-LOSS is given.

***OPENWD*** is the width of the opening perpendicular to the string of thermocouples.
***OPENHT*** is the height of the opening parallel to string of thermocouples.
***CTAMB*** is a constant ambient temperature.
***JTAMB*** is the channel number of the ambient temperature (NPDI [3]).

Note that if both CTAMB and JTAMB are left blank, the current default value for the ambient temperature will be used (20 °C, unless redefined).

*X* = ' X' - end-of-set mark.

**Input:** HCl2

**Variables:** *JTEMP(1), JTEMP(2), . . . , JTEMP(i) X* $1 \leq i \leq 30$
**Units:** temperature - °C
**Format:** EVALU8 (NPDI [2])

***JTEMP(i)*** is a single or average temperature channel number (NPDI [3]).
*X* = ' X' - end-of-set mark.

**Input:** HCl3

**Variables:** *TMPLOC(1), TMPLOC(2), . . . , TMPLOC(j) X* $2 \leq j \leq i$
**Units:** height - m
**Format:** EVALU8 (NPDI [2])

Enter this input only if 2 or more temperature channels were entered above.

***TMPLOC(j)*** is the height of the temperature reading of the above in the same order.
*X* = ' X' - end-of-set mark.

*Example*

```
VENT-LOSS
0.76 2.03 X
39   40  41  42  43  44  45  46  47  48  49  X
0.05 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 1.8 2.0 X
```

### 9.12 WT-RATE: Rate of Weight Loss

The subroutine makes two different calculations based on the total weight loss of an item: percent weight loss and rate of weight loss. Each of the calculations can be called individually. The calculated results are stored in their own channels created by the program. This subroutine does not reduce any raw data.

#### Percent Weight Loss

The percent weight loss is simply the total weight loss divided by the initial weight:

$$m = \frac{100\, m_\infty}{m_i} .$$

#### Rate of Weight Loss

The rate of weight loss is calculated by finding the least squares straight line fit of five adjacent points on the total weight loss curve. The rate of weight loss at the center point then, is simply the slope of the line:

$$\dot{m} = \frac{n\sum_{i=a}^{b} t_i m_i - \sum_{i=a}^{b} t_i \sum_{i=a}^{b} m_i}{n\sum_{i=a}^{b} t_i^2 - \left(\sum_{i=a}^{b} t_i\right)^2} .$$

For a five point running average (except near the end points), if $i$ is the point of interest, then $a = i - 2$ and $b = i + 2$. By dropping the "oldest" point and adding a new one, the slope can be found efficiently at every point.

Note that to be able to calculate the rate of weight loss, the corresponding time channel must be available.

**Input:** HCm1

**Variables:** *JCHAN, NPWL {,WEIGHT}, NRWL {JTIME} X*
**Units:** rate of weight loss - kg/s, time - s, weight - kg
**Format:** EVALU8 (NPDI [2])

Only one of these inputs is read each time the command WT-RATE is given.

*JCHAN* is the total weight loss channel number (NPDI [3]).
If *NPWL* is greater than zero, calculate percent weight loss. One channel created.
*WEIGHT* is the initial weight of object. Enter this value only if NPWL is greater than zero.
If *NRWL* is greater than zero, calculate rate of weight loss. One channel is created.
*JTIME* is the corresponding elapsed time channel. Enter this value only if NRWL is greater than zero.
*X* = ‘ X’ - end-of-set mark.

*Example*

```
WT-RATE
37 1 45.32 1 0 X
```

### 9.13 ZERO-TC: Radiation Correction for Gas Temperature Measurement

The subroutine finds the least squares fit of temperatures from various sized thermocouples and calculates the temperature for any other given size of thermocouple using the derived coefficients. Typically, the temperature is calculated for a zero diameter thermocouple.

The evaluated equation is simply a first-order linear equation of the form $T = a + bD$ where the temperature, $T$ is function of the thermocouple diameter D, and a and b are the derived coefficients from the least squares fit:

$$a = \frac{\sum_{i=1}^{n} D_i T_i - \frac{\sum_{i=1}^{n} T_i}{n}}{\sum_{i=1}^{n} D_i^2 - \frac{{\sum_{i=1}^{n} D_i}^{2}}{n}}, \qquad b = \frac{\sum_{i=1}^{n} T_i}{n} - a\frac{\sum_{i=1}^{n} D_i}{n} .$$

Note that at least two thermocouples must be input.

The calculated temperature is stored in a channel created by the program. This subroutine does not reduce any raw data. Any units may be used for the diameters and temperatures as long as they are consistent within a group.

**Input:** HCn1

**Variables:** *NGRP*
**Format:** open (NPDI [1])

Only one of these inputs is read each time the command ZERO-TC is given.

*NGRP* is the number of groups of thermocouples to be fit. Prepare NGRP sets of HCn2 inputs. (NGRP channels created one for each group.)

**Input:** HCn2

**Variables:** ***EVDIAM,] DIAM(1), TEMP(1), DIAM(2), TEMP(2), . . . , DIAM(i), TEMP(i) X*** ***1 ≤ i ≤ 25***
**Units:** diameter - consistent user selected units, temperature - °C
**Format:** EVALU8 (NPDI [2])

*EVDIAM* is the diameter for which the derived fit is evaluated (default, if omitted, is zero).
*DIAM(i)* is the thermocouple diameter.
*TEMP(i)* is the thermocouple channel number (NPDI [3]).
*X* = ' X' - end-of-set mark.

*Example*

```
ZERO-TC
1
0. 20. 39 15. 40 10. 41 5. 42 2. 43 X
```

## 10 References

[1] Smith, John M., Automatic data evaluation, manipulation, display, and plotting with SPEED, Computer Graphics, 4(2), 41-53; 1970 Fall.

[2] Peacock, R. D.; Smith, John M. SPEED2, A computer program for the reduction of data from automatic data acquisition systems. Natl. Bur. Stand. (U.S.) Tech. Note 1108; 1979 September.

[3] Smith, E. E., Fire Safety Evaluation of Rapid Transit Systems, Proceedings of the Eighth International Conference on Fire Safety, Millbrae, CA, January 17-21, 1983, pp 85-94 (1983).

[4] Kaufman, S.; Smith, E. E.; Przybyla, L. J.; and Williams, J. L., Large Scale Fire Tests of Building Riser Cables, Proceedings of the Eighth International Conference on Fire Safety, Millbrae, CA, January 17-21, 1983, pp 105-112 (1983).

[5] Parker, W. J., Calculations of the heat release rate by oxygen consumption for various applications. Natl. Bur. Stand. (U.S.) NBSIR 81-2427-1; 1982 March.

[6] Janssens, M., Measuring Rate of Heat Release by Oxygen Consumption, to be published in *Fire and Materials*.

[7] Peacock, R. D.; Davis, S.; Lee, B. T., An Experimental Data Set for the Accuracy Assessment of Room Fire Models, Natl. Bur. Stand. (U.S.) NBSIR 88-3752 (April 1988).

[8] Gőransson, U., Nordtest Full-scale Fire Tests, Proceedings of the Flame Retardants '90 Conference, London, UK, pp 34-42 (January 1990).

[9] Babrauskas, V.; Harris, R. H.; Gann, R. G.; Levin, B. C.; Lee, B. T.; Peacock, R. D.; Paabo, M.; Twilley, W.; Yoklavich, M. F.; Clark, H. M., Fire Hazard Comparison of Fire-Retarded and Non-Fire-Retarded Products, Natl. Bur. Stand. (U.S.) NBS Spec. Publ. 749 (July 1988).

[10] Babrauskas, V., Upholstered furniture room fires -- Measurements, comparison with furniture calorimeter data, and flashover predictions, *J. Fire Sci.* 4 (January/February 1984).

[11] Babrauskas, V., Development of the Cone Calorimeter -- A Bench Scale Heat Release Rate Apparatus Based on Oxygen Consumption, *Fire and Materials*. 8, 81-95 (1984).

[12] Karlson, B., Room Fires and Combustible Linings (Report SE-LUTVDG/TVBB-3050). Department of Fire Safety Engineering, Lund University, Lund, Sweden (1989).

[13] Cooper, L. Y.; Harkleroad, M.; Quintiere, J. G.; Rinkinen, W. J. An experimental study of upper hot layer stratification in full-scale multiroom fire scenarios. J. Heat Trans. 104: 741-749; 1982 November.

[14] Lee, B. T. Effect of ventilation on the rates of heat, smoke, and carbon monoxide production in a typical jail cell fire. Natl. Bur. Stand. (U. S.) NBSIR 82-2469; 1982 March.

[15] Lee, B. T. Effect of wall and room surfaces on the rates of heat, smoke, and carbon monoxide production in a park lodging bedroom fire. Natl. Bur. Stand. (U.S.) NBSIR 85-2998 1985 February.

[16] Steckler, K. D.; Quintiere, J. G.; Rinkinen, W. J., Flow induced by fire in a compartment. Natl. Bur. Stand. (U.S.) NBSIR 82-2520; 1982 September.

[17] Nakaya, I.; Tanaka, T.; Yoshida, M.; Steckler, K., Doorway flow induced by a propane fire. Fire Safety J. 10; 1986.

[18] Fang, J. B.; Breese, J. N., Fire development in residential basement rooms. Natl. Bur Stand. (U.S.) NBSIR 80-2120; 1980 October.

# A Notes on the Preparation of Data Inputs (NPDI) Read by RAPID

## A.1 NPDI [1] Open (List Directed) Formats

When entering values using an open (list directed) format, the value of the variable being entered must match the variable type (e.g., when entering an integer, a value with a decimal point must not be found). Therefore, variable names used in this program follow the standard FORTRAN convention for typing:

> unless otherwise noted -
>
> > variable names beginning with the letters A through H or O through Z are real;
> > variable names beginning with the letters I through N are integer.

## A.2 NPDI [2] Special Format Read by Subroutine EVALU8

Subroutine EVALU8 is a general purpose data input reading routine. It is called by many of the data reduction subroutines. It reads and counts the number of values on the input, stores them and their types (integer, real, or character) in data arrays, and returns the data and control to the calling subroutine.

An input value is defined as the string of digits and/or characters found between spaces or commas on a data input. The end of the set of values is signaled by the characters ‘ X’ (a space followed by an X) after the last value. When no limit is imposed by the calling subroutine, up to 100 values may be read in one set. If all the values cannot fit on one input, they may continue on to the next input. Inputs will continue to be read until ‘ X’ or an illegal character is encountered. If no digits or characters are encountered before the first comma or between commas, a real value of 0.0 is assumed.

The legal inputs are as follows:

| | |
|---|---|
| 0 1 2 3 4 5 6 7 8 9 | |
| + | (plus) |
| - | (minus) |
| . | (decimal point) |
| E | (exponent) |
| Z | (“zero”) |
| R | (“range” or “span”) |
| $ | (dollar sign) |
| , | (comma) |
| | (space) |
| | (a character string enclosed in quotation marks such as ‘CO2’) |
| X | (end-of-set mark) |

This subroutine will accept integers, reals, or character strings with the stipulation that the values must match the type of the variable being entered. (See NPDI [1] above.) A real value must include a decimal point or "E."

The letters Z and R stand for "zero" and "range" (or "span") respectively. If the "zero" and "span" scans have been identified through the use of Part F, these letters represent the values stored by the program as representing the smallest value from the "zero" scans and the largest value from the "span" scans for each instrument (column) in the data matrix. Only certain routines may use these values, therefore, be sure to read the individual instructions carefully.

The dollar sign ($) also has a special meaning to this subroutine. It signals that a channel created during execution of the program is to be used. The number following the dollar sign represents the order in which the channel was created. The dollar sign and the digits are automatically replaced by the channel number in which the values are stored. See NPDI [3], below.

Note that the format described here is almost identical to "open" (list directed) format. The difference is the limit on the number of values that can be read and the special characters: the dollar sign ($) and the letters X, Z and R.

### A.3 NPDI [3] Channels Created by the Program

This program creates new channels in which to store some of the calculated data. To identify a created channel for use by a data reduction subroutine or by the plot routine (data input I1), only the order in which the channels were created need be known. For example, to plot the data stored in the fifth channel created, enter the number '$5' where the channel number is normally entered. The program automatically replaces the dollar sign ($) and the 5 with the proper channel number. This method of identifying a channel may be used in any of the subroutines when the input data are read using subroutine EVALU8. (See NPDI [2] above.) It may also be used when entering channel numbers for plots or as noted in other specific command instructions.

The dollar sign method of identifying created channels can not be used if the channel was created in a previous execution of the program. [In those cases, the channel must be handled in the normal manner, appearing in the instrument list (data input D3) and using the complete, right-justified, six-digit number.]

The created channel numbers can be assigned for use in two ways: automatically by the program and by the user by means of the SPECIFY command. If the program chooses the channel number, the number is chosen such that the smallest available channel number from an unused series of channels is used. (Note: a channel series is a value from 0 to 9 and is identified by the first digit in the channel number (see NPDI [5] below).) The program checks the original instrument list and determines the series which have channels used. The remaining series are put in the created channel pool for use as needed. The total number of channels that can be automatically drawn from the pool is then the number of empty series times 100. Also, for that reason, it is wise to choose a channel number for time in a series that is already being used.

### A.4 NPDI [4] Saving Unreduced (Raw) Data in a Formatted Data Matrix

The option for saving raw, unreduced data in a formatted data matrix is input on Data Input C1 (Variable INSAVE). By setting INSAVE to the appropriate value the first time the raw data set is used, a formatted raw data set can be saved. Since this new data set is in a data-collection-system-independent format, the program can (in subsequent executions) read the input data without having to interpret for format correctness (i.e., read numbers instead of reading character by character), which is significantly faster.

### A.5 NPDI [5] Instrument Numbers - Management of the Test Number Prefix

An instrument number is a six digit number whose first three digits represent a test number and whose last three digits represent a channel number. There are two variables, INTEST (Data Input C1) and NTEST (Data Input C2), that control the test number prefix part of the instrument numbers. INTEST controls the prefix during input, NTEST controls the prefix during output (when NPNCH (Data Input C2) is greater than zero).

INTEST can direct the program to either pass the test number through or to strip off the test number, leaving only the channel number (effectively it changes the test number prefix to 000, which is insignificant). If the test number is passed through and it is significant, it must be used when identifying instruments for data transformations and plotting.

NTEST can direct the program to pass the test number "as is" to the saved reduced data file or to replace the test number with a value between 1 and 999 inclusive. NTEST has no effect if NPNCH is zero, since no reduced data are saved.

### A.6 NPDI [6] Output of the Transformed Data Matrix

The option for printing (or not printing) the transformed data matrix is input on Data Input C2. In the SPEED2 version, the matrix is either not printed (NPRT.EQ.0) or printed (NPRT.NE.0). The RAPID version is slightly different in that it prepares a table of minimums, maximums and averages for each channel and, consequently, the options for NPRT have also changed:

NPRT $= 0$, neither data matrix nor summary table is printed
$< 0$, only the summary table is printed
$> 0$, both the matrix and the summary table are printed.

This change is compatible with earlier versions of the program.

### A.7 NPDI [7] Constant Value Conversion Inputs

In subroutine CONV, after all other conversions and calculations are done, the data undergo one final reduction before finally leaving the subroutine. In the SPEED2 version, this reduction is linear and of the form

REED = (REED * C) + ADD

where C and ADD are constants entered with data input D3. The RAPID version is different in that it includes an exponential value:

REED = ((REED * C) + ADD) ** POWER

where C, ADD, and POWER are the constants entered. Data input D3 is modified as follows to accommodate the change:

| | |
|---|---|
| Variables: | C(i),ADD(i),POWER(i) |
| Format: | 1X,3F15.6 |

The following default conditions exist to help avoid errors: If ABS(POWER) < 1., (REED * C) < 0 and ADD < 0, then the calculated value defaults to zero. If POWER = 0., POWER defaults to 1. The automatically assigned constant values for channels created by the program are:

C = 1.0     ADD = 0.0     POWER = 1.0

This change is compatible with earlier versions of the program.

### A.8 NPDI [8] Variables in Brackets ([]) and Braces ({})

In the documentation above, some of the variables in the variable lists under each command appear within brackets ([]) or braces ({}). The variables within brackets are optional; the variables with braces are conditional. The options and/or conditions are specified to the right under the Comments. If the variable is not used as an argument, the comma used to separate it from the other variables (if used) should also be omitted.

NIST-114A
(REV. 3-89)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

# BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NIST/SP-798

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
January 1991

4. TITLE AND SUBTITLE

A Users Guide for RAPID, Version 2.3

5. AUTHOR(S)

Richard D. Peacock, J. Newton Breese, C. Lynn Forney

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

U.S. Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899

10. SUPPLEMENTARY NOTES
Supersedes NBS SP 722

□ DOCUMENT DESCRIBES A COMPUTER PROGRAM; SF-185, FIPS SOFTWARE SUMMARY, IS ATTACHED.

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

The voluminous amount of data that can be collected by automatic data acquisition systems during large scale fire tests requires the use of a digital computer for the reduction of data. RAPID is a stand-alone program specifically designed to convert raw instrument voltages collected during such tests into a meaningful form. The reduced data can also be used alone or in combinations to obtain derived quantities. The program is written with the ability to accept data from a user defined data acquisition system and to check the correctness of data being analyzed. The data can be converted into meaningful scientific units and then presented in tabular or printer plot form or stored for future processing. This guide provides detailed instructions for the use of the program.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

Computer program; data acquisition; data reduction; fire tests

13. AVAILABILITY

☒ UNLIMITED
□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).
☒ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.
☒ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
112

15. PRICE

ELECTRONIC FORM

☆ U.S. GOVERNMENT PRINTING OFFICE: 1991 — 281-557 / 40401

# NIST Technical Publications

## Periodical

**Journal of Research of the National Institute of Standards and Technology**—Reports NIST research and development in those disciplines of the physical and engineering sciences in which the Institute is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Institute's technical and scientific programs. Issued six times a year.

## Nonperiodicals

**Monographs**—Major contributions to the technical literature on various subjects related to the Institute's scientific and technical activities.

**Handbooks**—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

**Special Publications**—Include proceedings of conferences sponsored by NIST, NIST annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

**Applied Mathematics Series**—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

**National Standard Reference Data Series**—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NIST under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NIST by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW., Washington, DC 20056.

**Building Science Series**—Disseminates technical information developed at the Institute on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

**Technical Notes**—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NIST under the sponsorship of other government agencies.

**Voluntary Product Standards**—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NIST administers this program as a supplement to the activities of the private sector standardizing organizations.

**Consumer Information Series**—Practical information, based on NIST research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

*Order the* **above** *NIST publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

*Order the* **following** *NIST publications—FIPS and NISTIRs—from the National Technical Information Service, Springfield, VA 22161.*

**Federal Information Processing Standards Publications (FIPS PUB)**—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NIST pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

**NIST Interagency Reports (NISTIR)**—A special series of interim or final reports on work performed by NIST for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

**U.S. Department of Commerce**
National Institute of Standards and Technology
(formerly National Bureau of Standards)
Gaithersburg, MD 20899