LA-SUB--96-18

Final Report
LANL Subcontract 9-XH2-Y3109-1
NMSU account 01-4-23426

Dr. Edward Hensel, P.E.
Mechanical Engineering, Department 3450
New Mexico State University
Las Cruces, NM 88003-0001

June 1993

Contents

1 Introduction

This report contains thefinal budget report, users guides, programmer's guide, and copies of preliminary letter progress reports related to the graphics post-processing support for the PC versions of the FIRIN and FIRAC computer simulation programs. All required Los Alamos Certification forms are prepared by the NMSU Office of Grants and Contracts, and are provided separate from this document.

All computer programs developed and modified under this contract have been submitted to Dr. Bill Gregory at Los Alamos National Laboratories, on a 3.5" DOS format floppy disk. Inquiries regarding obtaining the source code should be directed to Dr. Gregory.

FIRINPC and FIRACPC Graphics Post-Processor Support User's Guide and Programmer's Reference

Dr. Edward Hensel, P.E.
Mechanical Engineering, Department 3450
New Mexico State University
Las Cruces, NM 88003-0001

March 1992

Contents

Chapter 1

User's Guide

1.1 Introduction

FIRIN is a computer program used by DOE fire protection engineers to simulate hypothetical fire accidents in compartments at DOE facilities. The FIRIN code is typically used in conjunction with a ventilation system code such as FIRAC, which models the impact of the fire compartment upon the rest of the system. The code described here, FIRINPC is a PC based implementation of the full mainframe code FIRIN. In addition, FIRINPC contains graphics support for monitoring the progress of the simulation during execution and for reviewing the complete results of the simulation upon completion of the run. This document describes how to install, test, and subsequently use the code FIRINPC, and addresses differences in usage between the PC version of the code and its mainframe predecessor. The PC version contains all of the modeling capabilities of the earlier version, with additional graphics support. This user's guide is a supplement to the original FIRIN report published by the NRC.

FIRAC is a computer program used by DOE fire protection engineers to simulate the transient response of complete ventilation systems to fire induced transients. FIRAC has the ability to use the FIRIN code as the driving function or source term for the ventilation system response. The current version of FIRAC does not contain interactive graphics capabilities.

A third program, called POST, is made available for reviewing the results of a previous FIRIN or FIRAC simulation, without having to recompute the numerical simulation. POST uses the output data files created by FIRINPC and FIRACPC to avoid recomputation.

1.2 PC Requirements

You must use an IBM PC or fully compatible alternative, with a VGA graphics board and monitor. You should have a math coprocessor, and preferably use an 80386 or greater machine. If you wish to change the source code or recompile it as described in the programmer's reference, you will need to have a license for MICROSOFT FORTRAN Version 5.x. While the code can be run from a floppy disk, it is STRONGLY recommended that you use a hard disk.

1.3 Installation

This section of the user's guide contains instructions on how to install the FIRINPC and POST codes on your PC. In addition, we will discuss how to install any necessary data files, and run through a debugging test case, corresponding to sample problem number 1 from the original FIRIN manual.

Three files are needed to work with FIRINPC. The executable codes are contained in FIRINPC.EXE and POST.EXE. These two executable codes in turn rely upon a Microsoft FORTRAN Version 5.x font file named COURB.FON. The font file is provided on the distribution disk for users who may not have a Microsoft FORTRAN compiler on their PC. If you do indeed have a Microsoft FORTRAN Version 5.x compiler on your PC, with the INCLUDE and LIB paths properly declared in your AUTOEXEC.BAT file, then you may not need the font file.

First let us assume that your distribution diskette is to be placed in drive a:, and that your hard disk is designated as drive c:. To install the system, issue the following commands:

```
c:> cd c:\
c:> mkdir FIRINPC
c:> cd FIRINPC
c:\FIRINPC> copy a:*.exe *.*
c:\FIRINPC> copy a:*.fon *.*
c:\FIRINPC> copy a:*.dat *.*
```

The first command moves your current directory to the hard disk. Next, we create a new directory to receive the codes and data files, then move into that directory. We complete the installation by copying the executable files, the font file, and the sample problem 1 data files to your hard disk. At this point, you should be ready to begin work and validation of the installation.

1.4 Post Processing FIRIN Graphics

Now that we have copied the files onto the hard disk, we need to verify the installation and learn how to use the programs. Let's start with the post-processing program, POST. Enter the command

```
c:\FIRINPC> POST
```

The program will begin execution by presenting the first level menu to you. From this menu, you may select screen graphics of the FIRIN or FIRAC responses, or you may elect to generate tektronix-compatable hard copy files which may be later printed on an LNO3-Plus laser printer. To begin, select option 1 from the menu:

The program will next read the output data files from sample problem 1 provided with the distribution disk. After reading the data, POST will display a menu of plot options, as shown below.

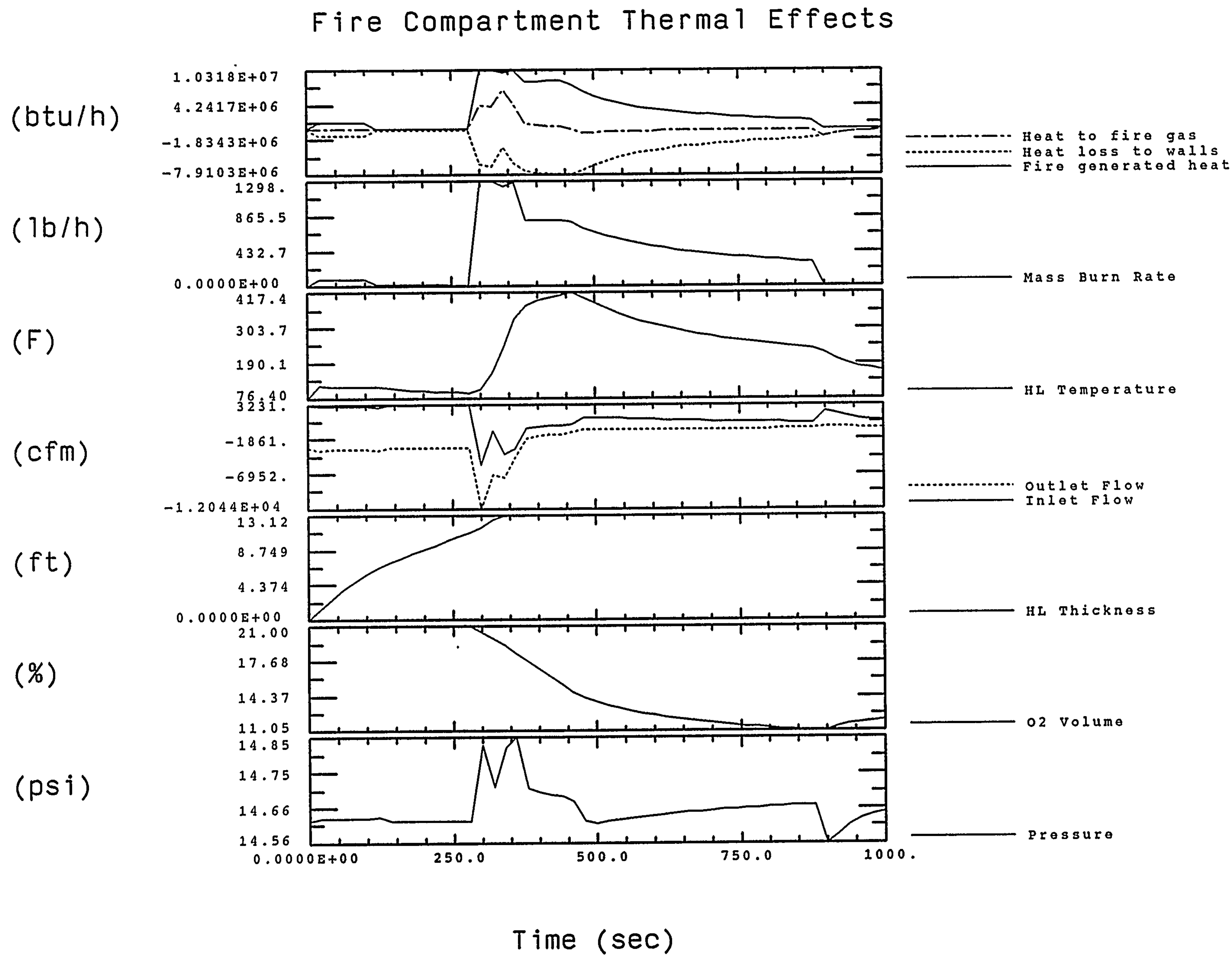
```
Which option would you like?
1)  FIRIN Fire Compartment Graphics Review
2)  FIRAC Ventilating System Graphics Review
3)  FIRIN Fire Compartment Hard Copies
4)  FIRAC Ventilating System Hard Copies
99) Exit to DOS ............................ 1
```

```
Which kind of graphics would you like?

 1) Fire Compartment Thermal Effects
    11) Thermal Energy Balance              15) Hot Layer Thickness
    12) Mass Burn Rate                      16) Oxygen Volume Fraction
    13) Hot Layer Temperature               17) Fire Compartment Pressure
    14) Volumetric Flow Balance

 2) Smoke Source Term vs. Time           3) Radioactive Source Term vs. Time
    21) Smoke Accumulation                  31) Radioactive Source Accumulation
    22) Hot Layer Loss Summary              32) Hot Layer Loss Summary
    23) Hot Layer Mass Balance              33) Hot Layer Mass Balance
    24) Cold Layer Loss Summary             34) Cold Layer Loss Summary
    25) Cold Layer Mass Balance             35) Cold Layer Mass Balance

 4) Particulate Size Distribution
    41) Radioactive Hot Layer               43) Smoke Hot Layer
    42) Radioactive Cold Layer              44) Smoke Cold Layer

99) RETURN to main menu.
```

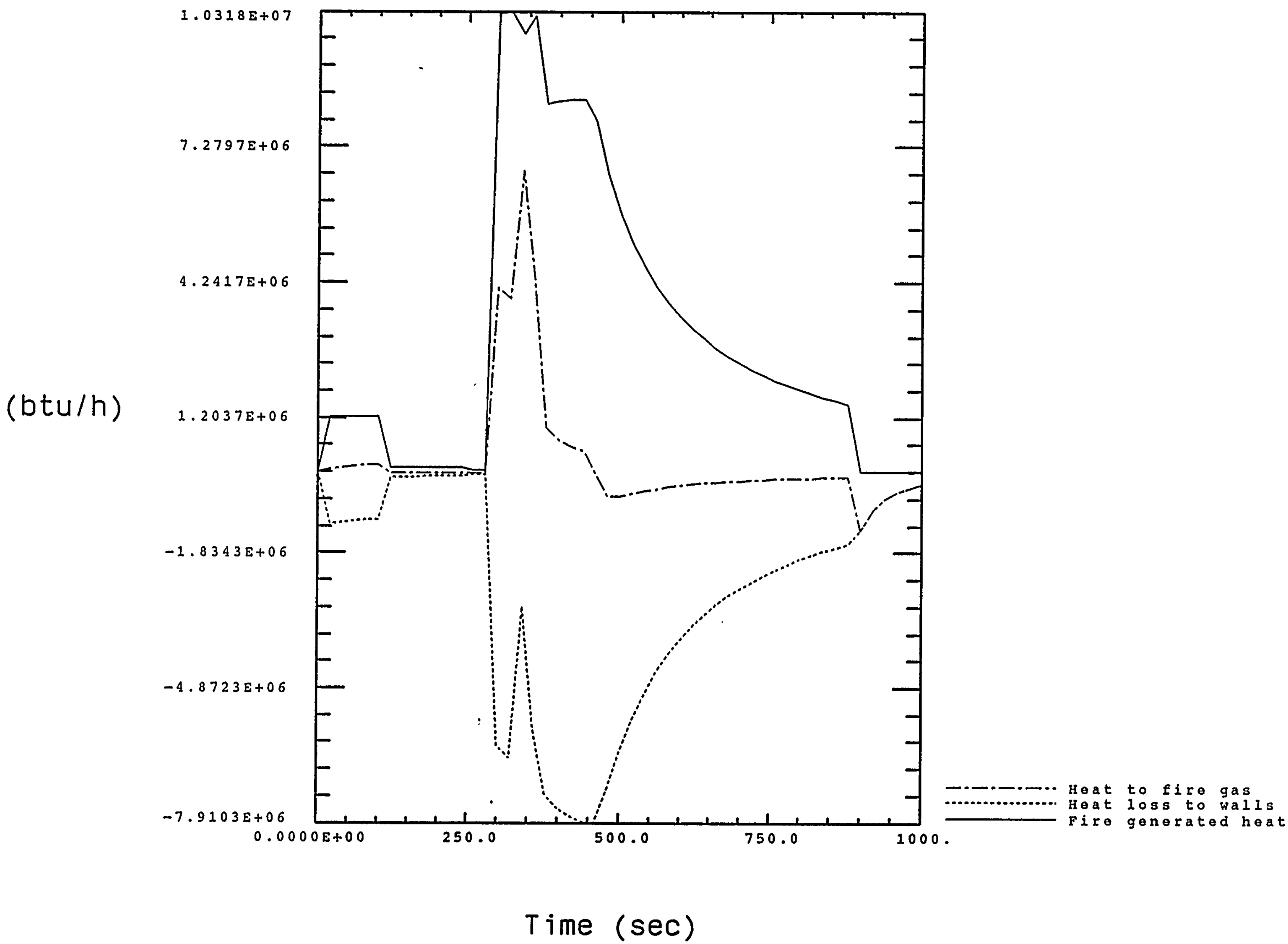
You may now choose from among the 25 plotting options. Let us first view plot (1), which contains a summary of the fire compartment thermodynamics. To begin, type "1" and press the enter key on your keyboard. The program will now display a sequence of seven plots simultaneously on the screen. These seven plots correspond to each of the individual plot options labeled (11) through (17). You may review the mass flows, fire compartment pressure, temperatures, oxygen content, and energy balance for sample problem number 1 from the FIRIN manual. After you have finished looking at the plots, press the enter key to return to the main menu. If you next type "11" and press enter, you will see a full screen version of the thermal energy balance plot viewed previously in the fire compartment summary plot. The full screen version is more convenient for studying details of a particular response, while the summary plots are useful for understanding the relationships between events occurring in the fire compartment. After viewing plot (11), press the enter key again to return to the main option menu. To complete the installation verification of `POST`, you may review a combination of the other plotting options. After you finish viewing each plot, remember to press the enter key to return back to the main menu. To exit the post processing program, type "99" and press the enter key one more time.

This is precisely the same procedure you can use in the future to review the results of FIRIN simulations for your own applications. Any time you wish to use the post processor, simply enter the command "POST" at the DOS prompt.

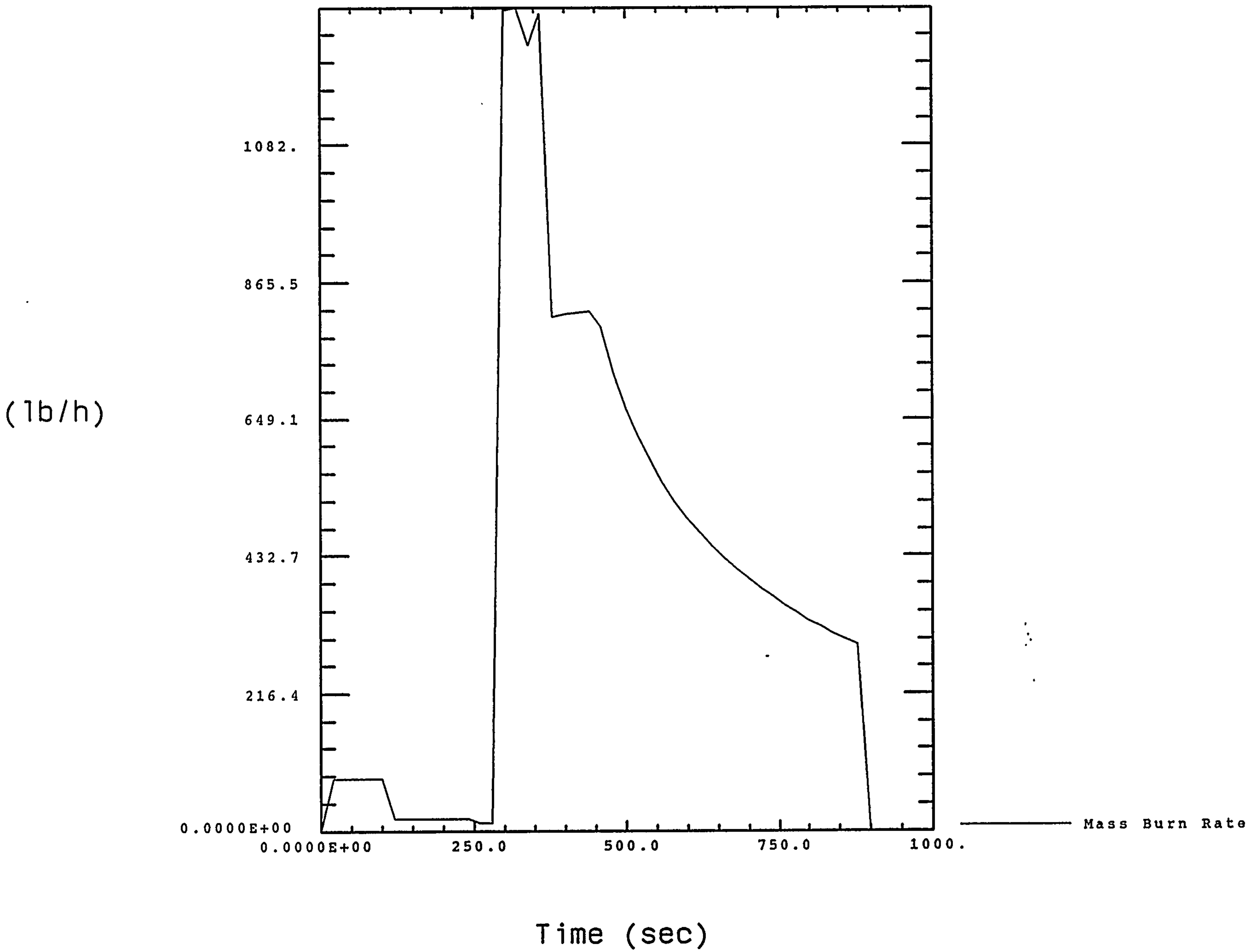
Some typical plotting results of from FIRIN sample problem number 1 are given on the following pages.


Fire Compartment Thermal Effects
(btu/h)
1.0318E+07
4.2417E+06
-1.8343E+06
-7.9103E+06
Heat to fire gas
Heat loss to walls
Fire generated heat
(lb/h)
1298.
865.5
432.7
0.0000E+00
Mass Burn Rate
(F)
417.4
303.7
190.1
76.40
HL Temperature
(cfm)
3231.
-1861.
-6952.
-1.2044E+04
Outlet Flow
Inlet Flow
(ft)
13.12
8.749
4.374
0.0000E+00
HL Thickness
(%)
21.00
17.68
14.37
11.05
O2 Volume
(psi)
14.85
14.75
14.66
14.56
Pressure
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)

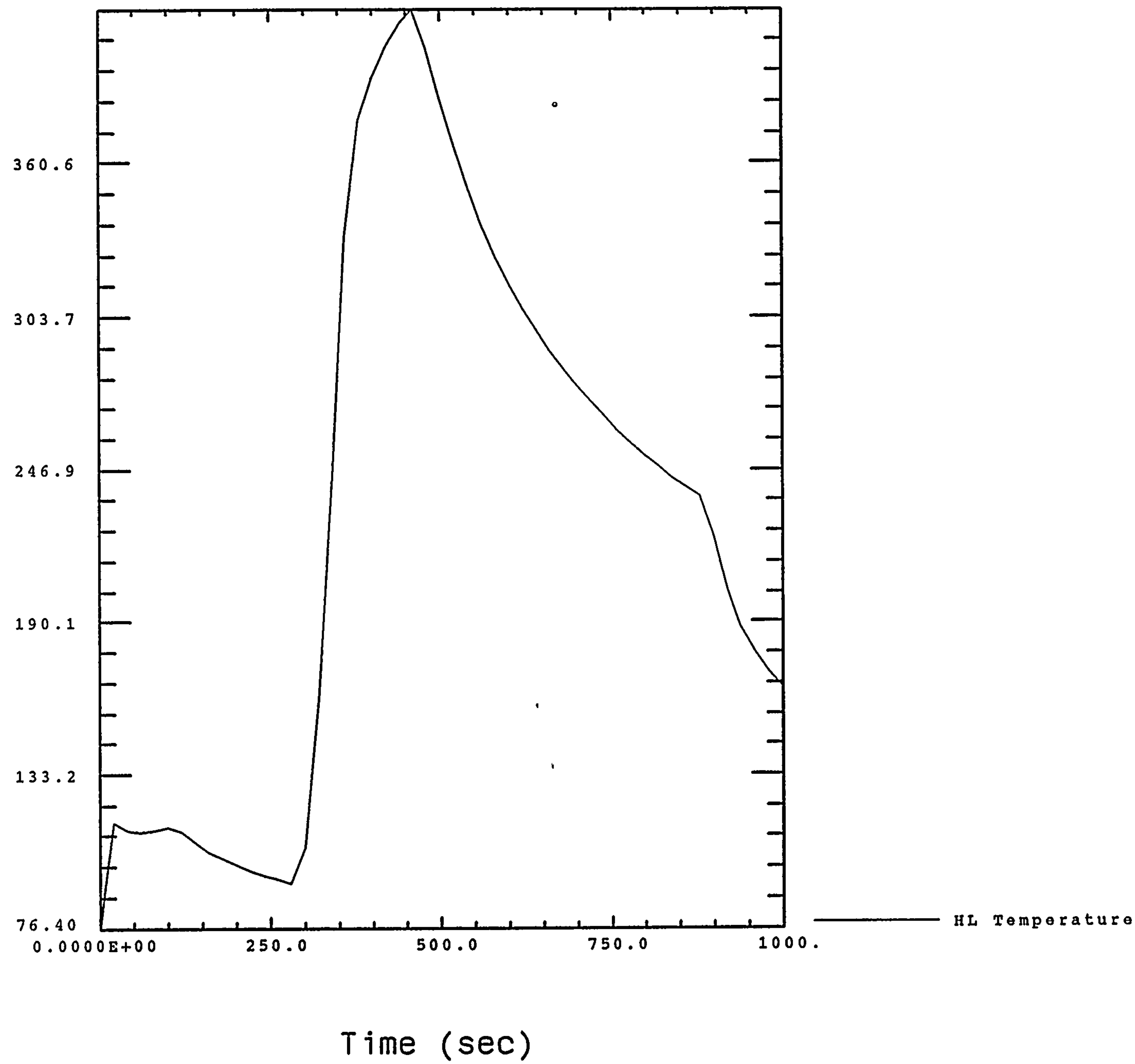
Fire Compartment Thermal Effects



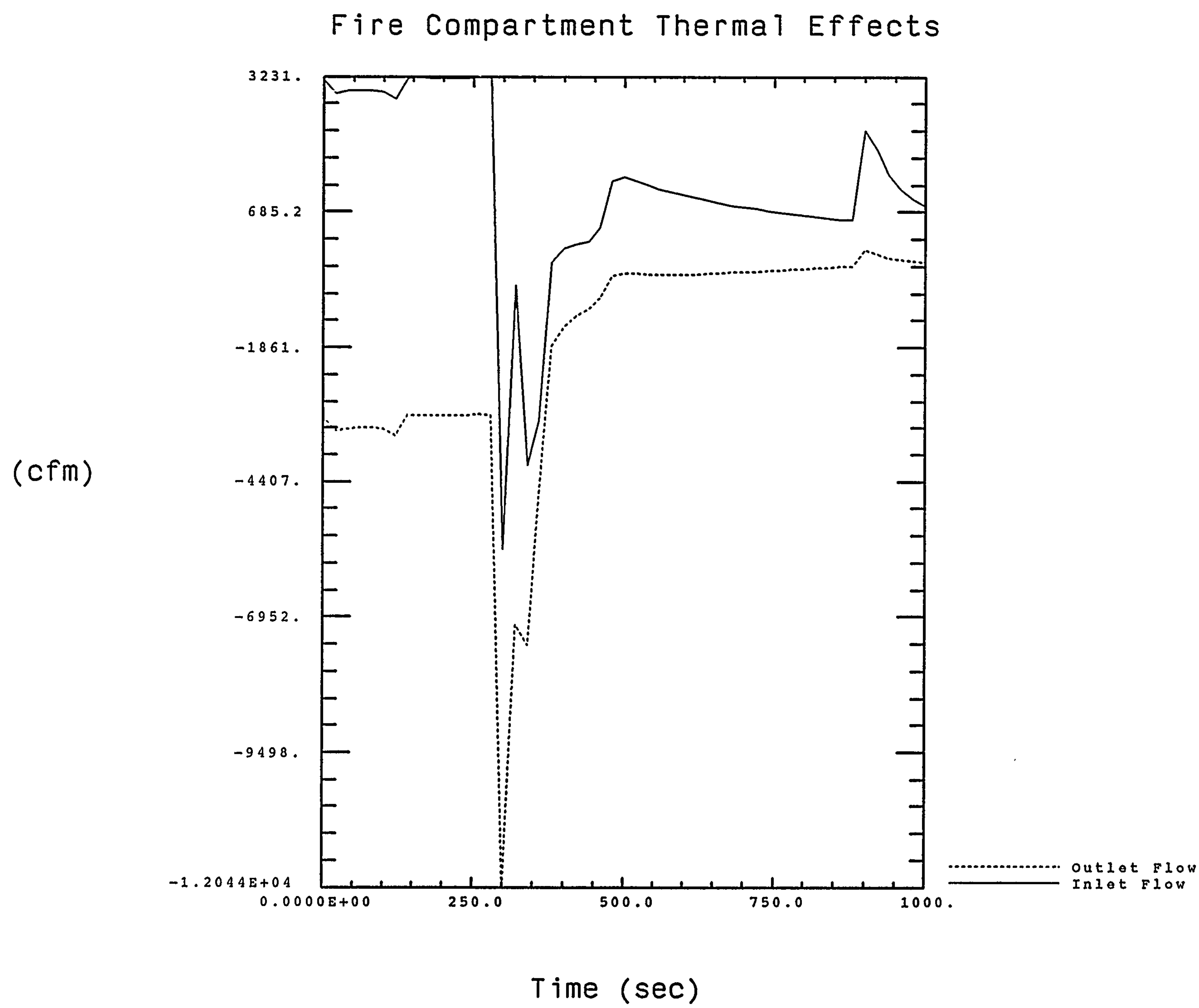
Fire Compartment Thermal Effects



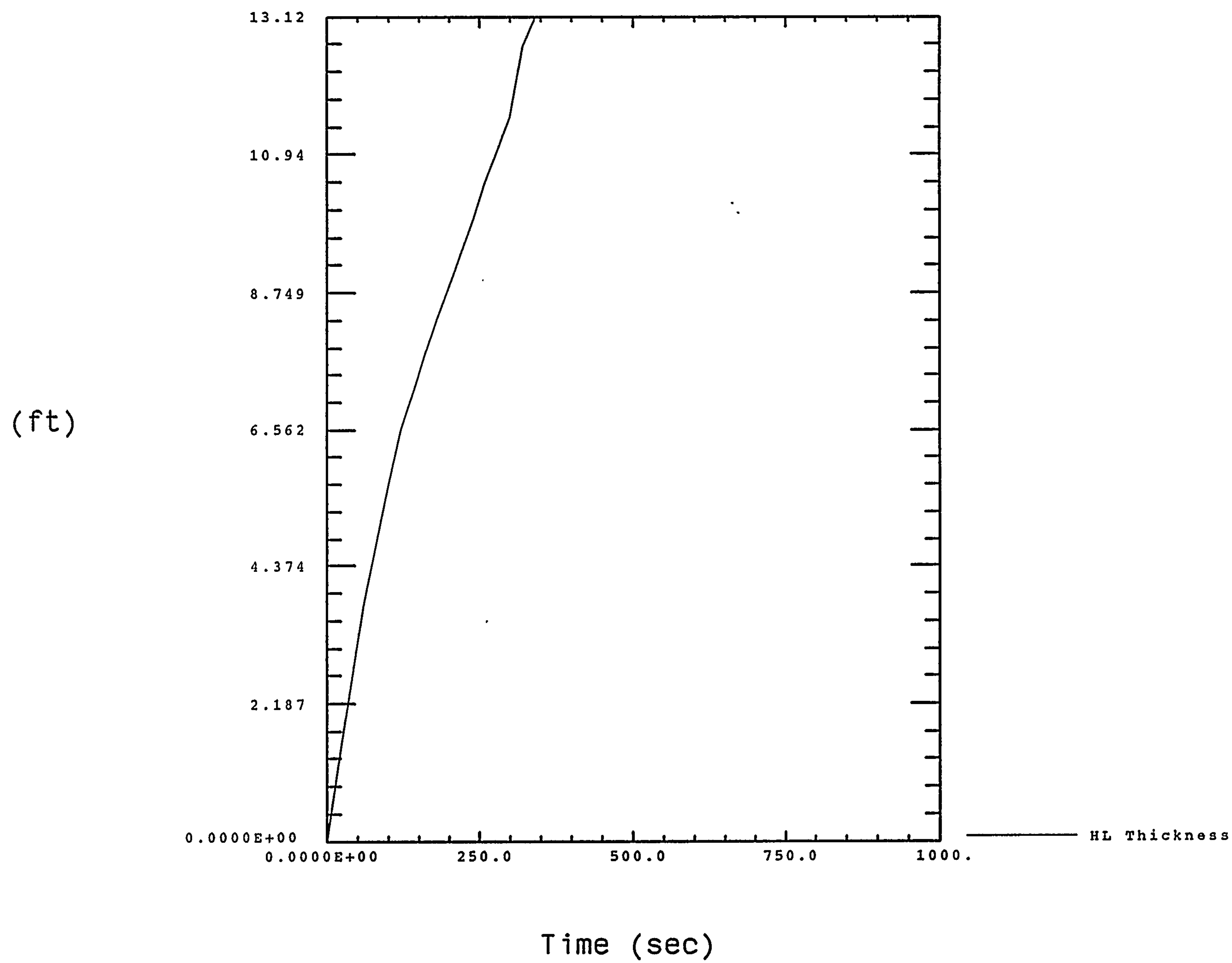
Fire Compartment Thermal Effects

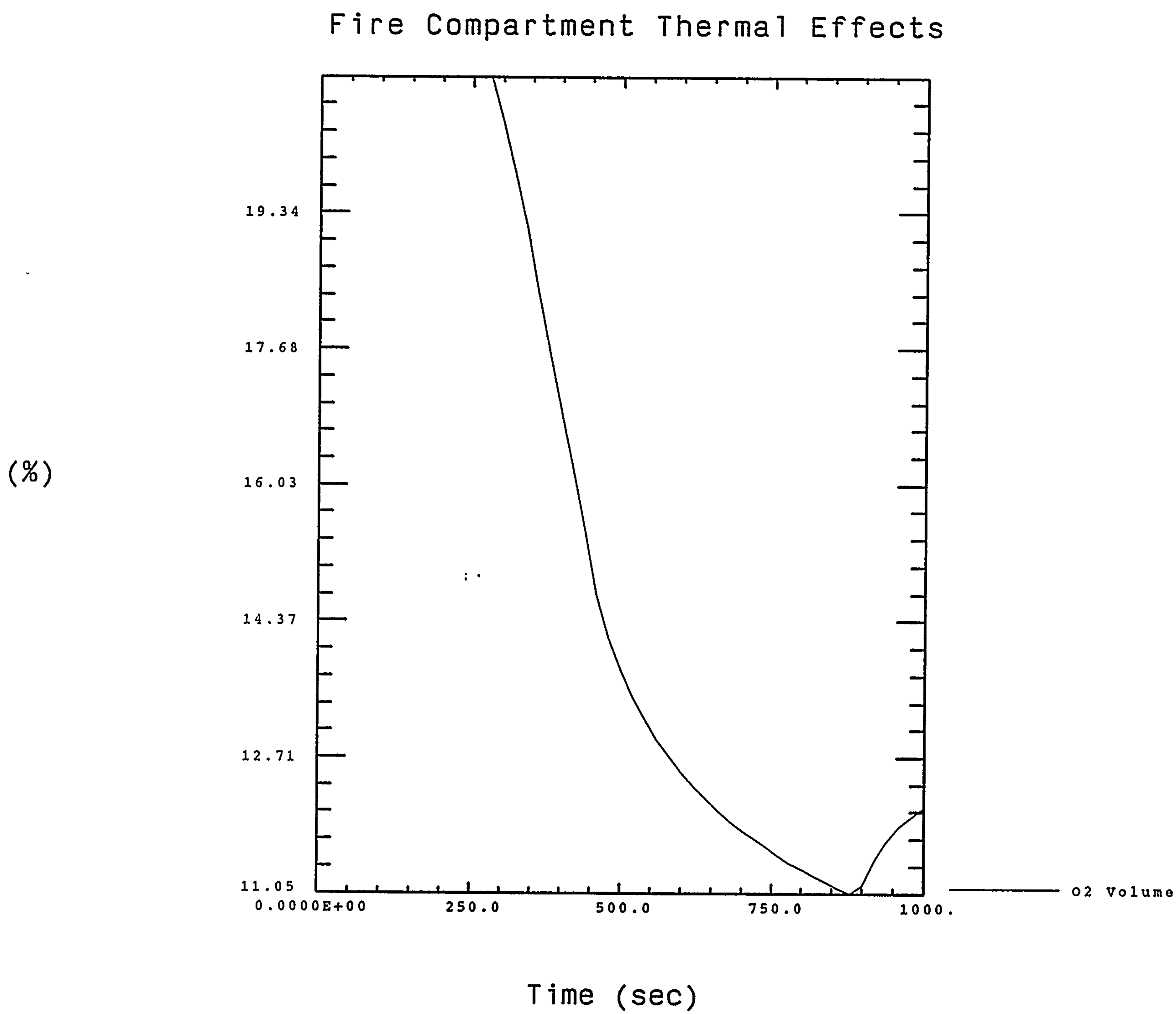


(F)

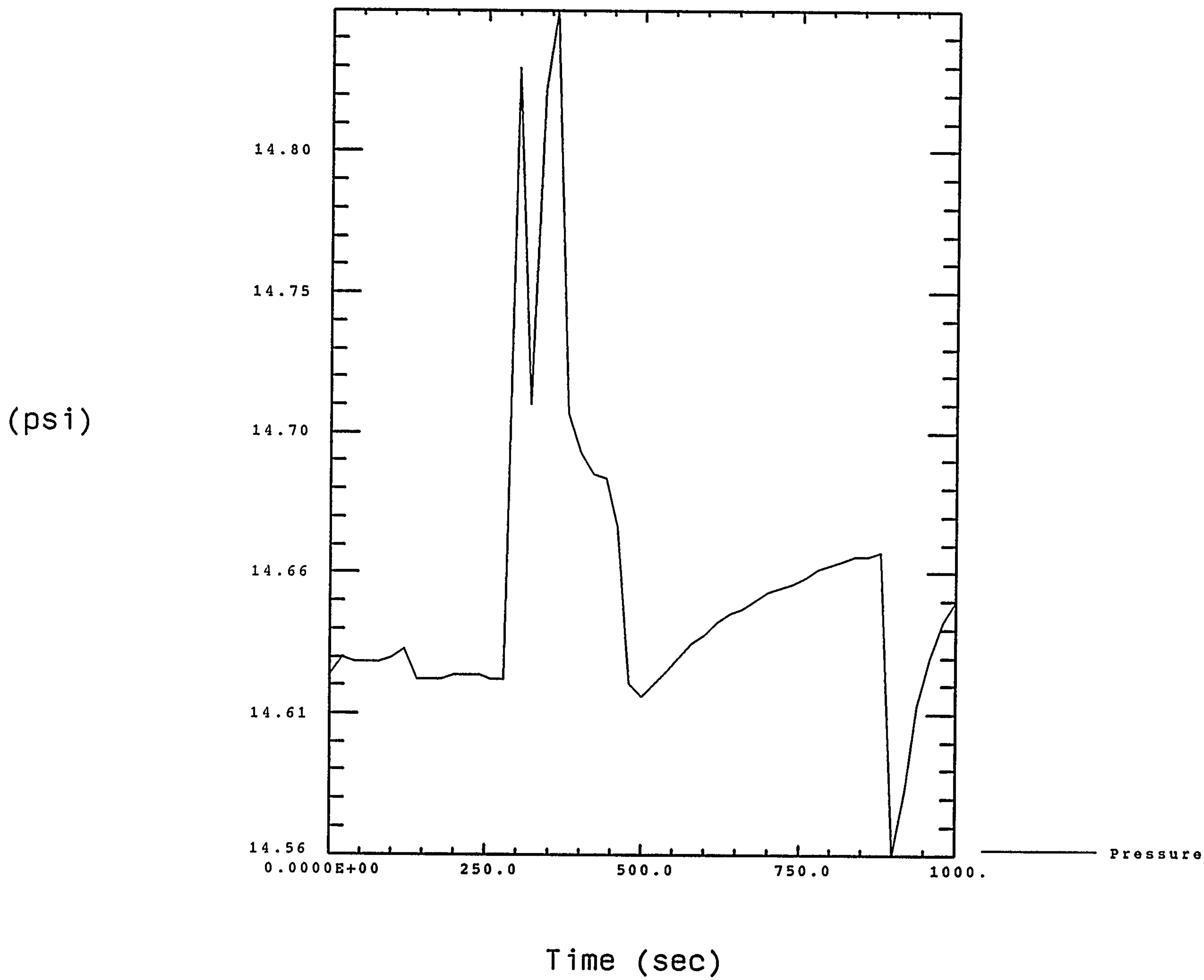

Fire Compartment Thermal Effects
3231.
685.2
-1861.
-4407.
-6952.
-9498.
-1.2044E+04
(cfm)
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Outlet Flow
Inlet Flow

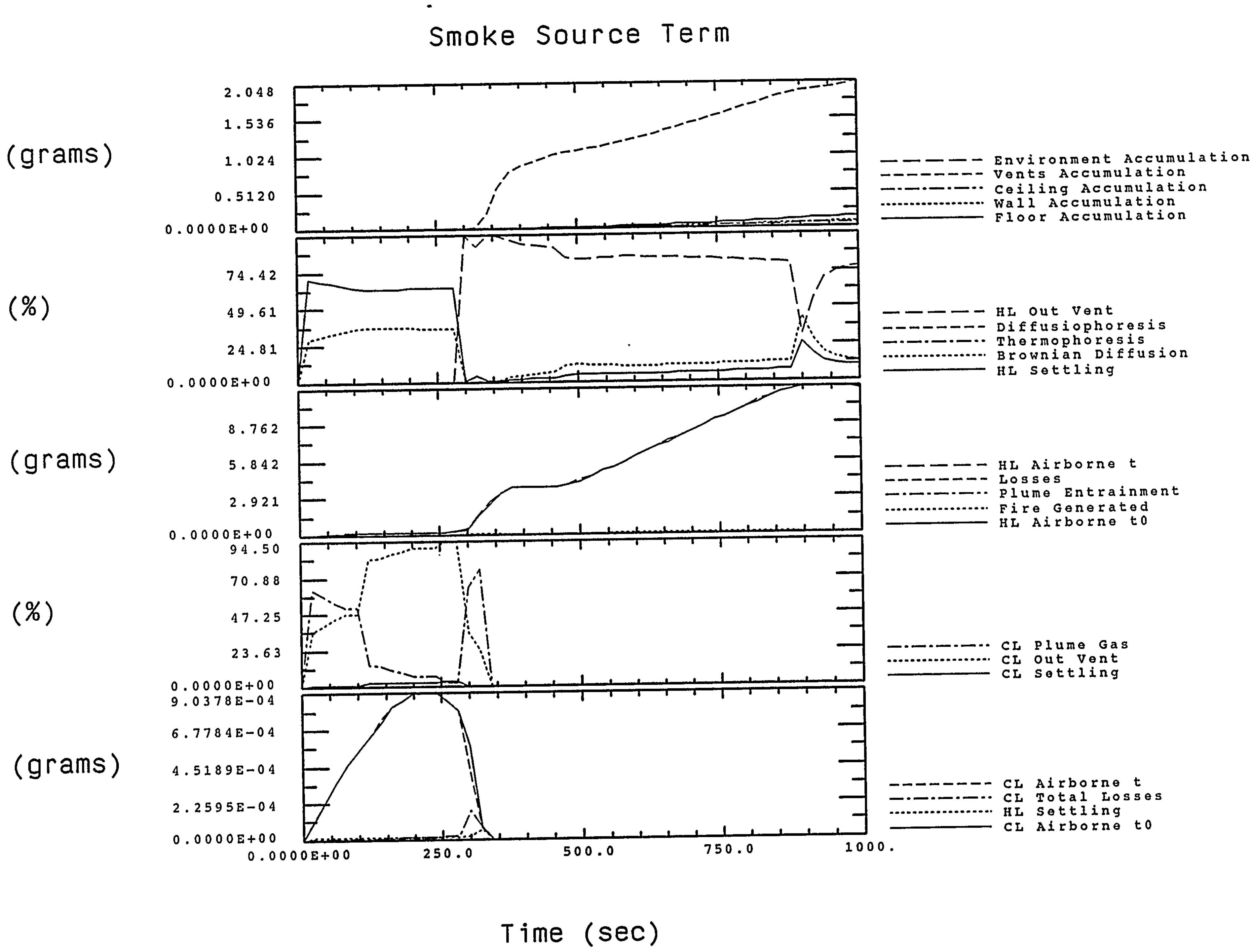
Fire Compartment Thermal Effects

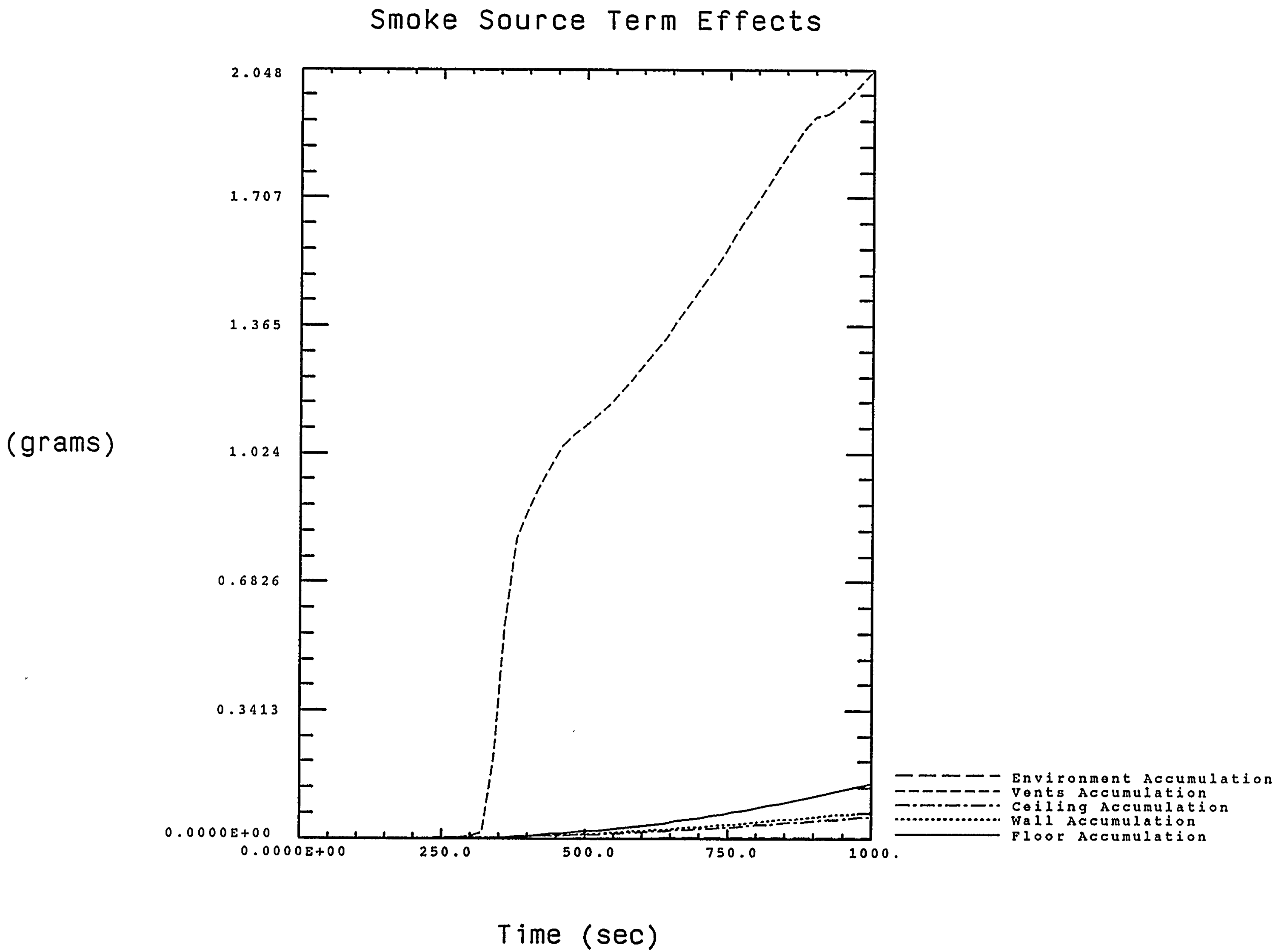


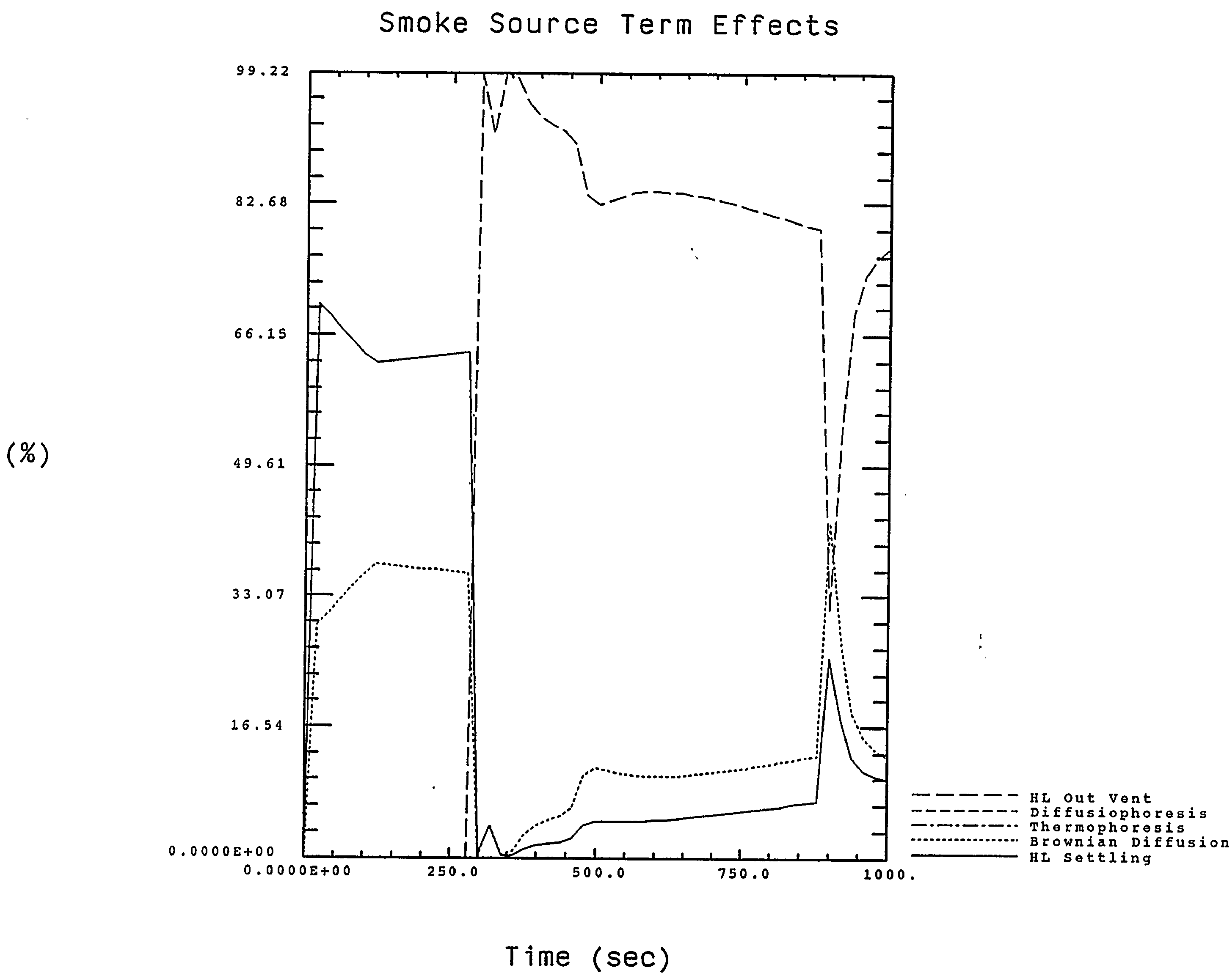

Fire Compartment Thermal Effects
(%)
19.34
17.68
16.03
14.37
12.71
11.05
0.0000E+00
250.0
500.0
750.0
1000.
O2 Volume
Time (sec)

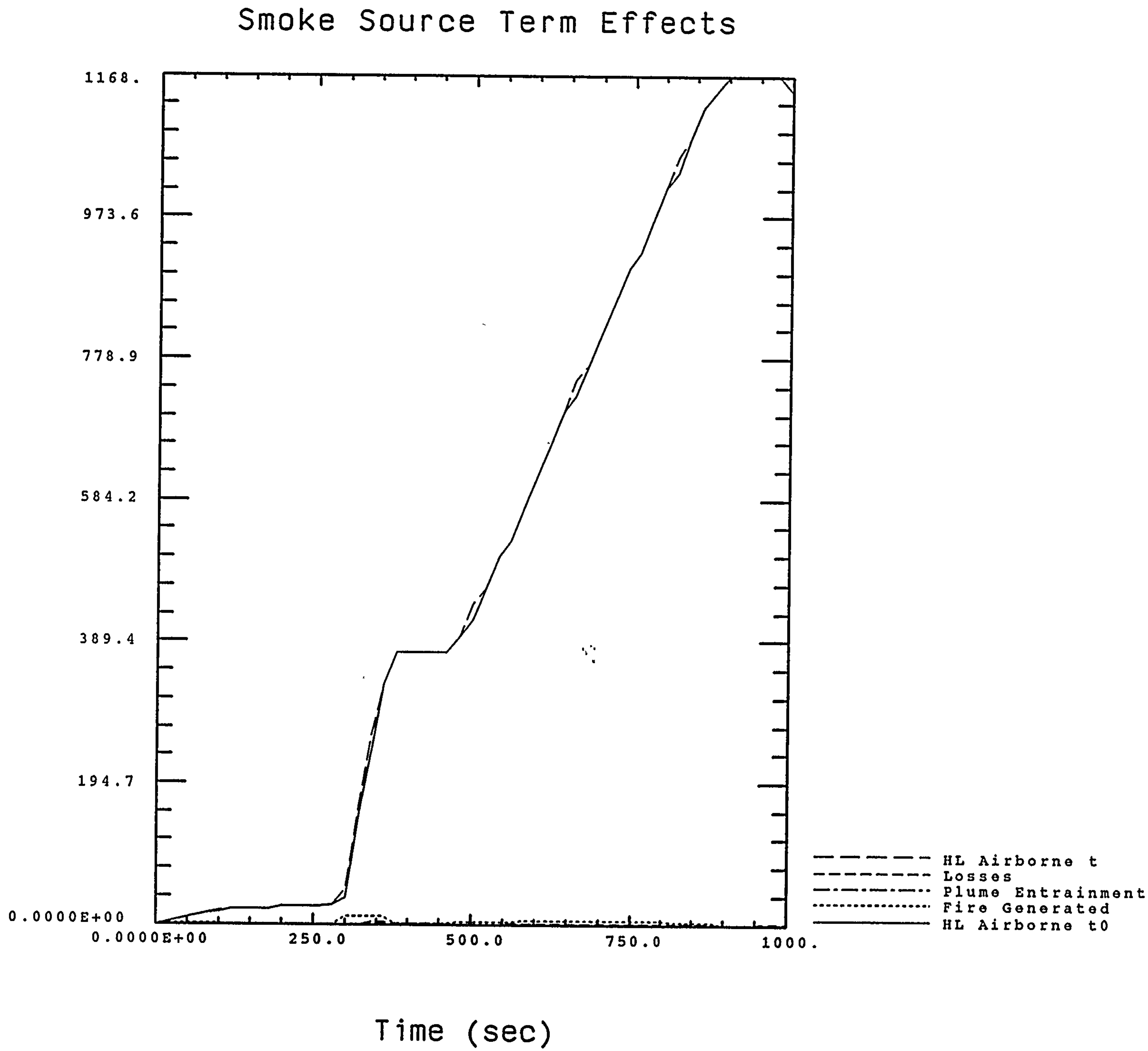
Fire Compartment Thermal Effects

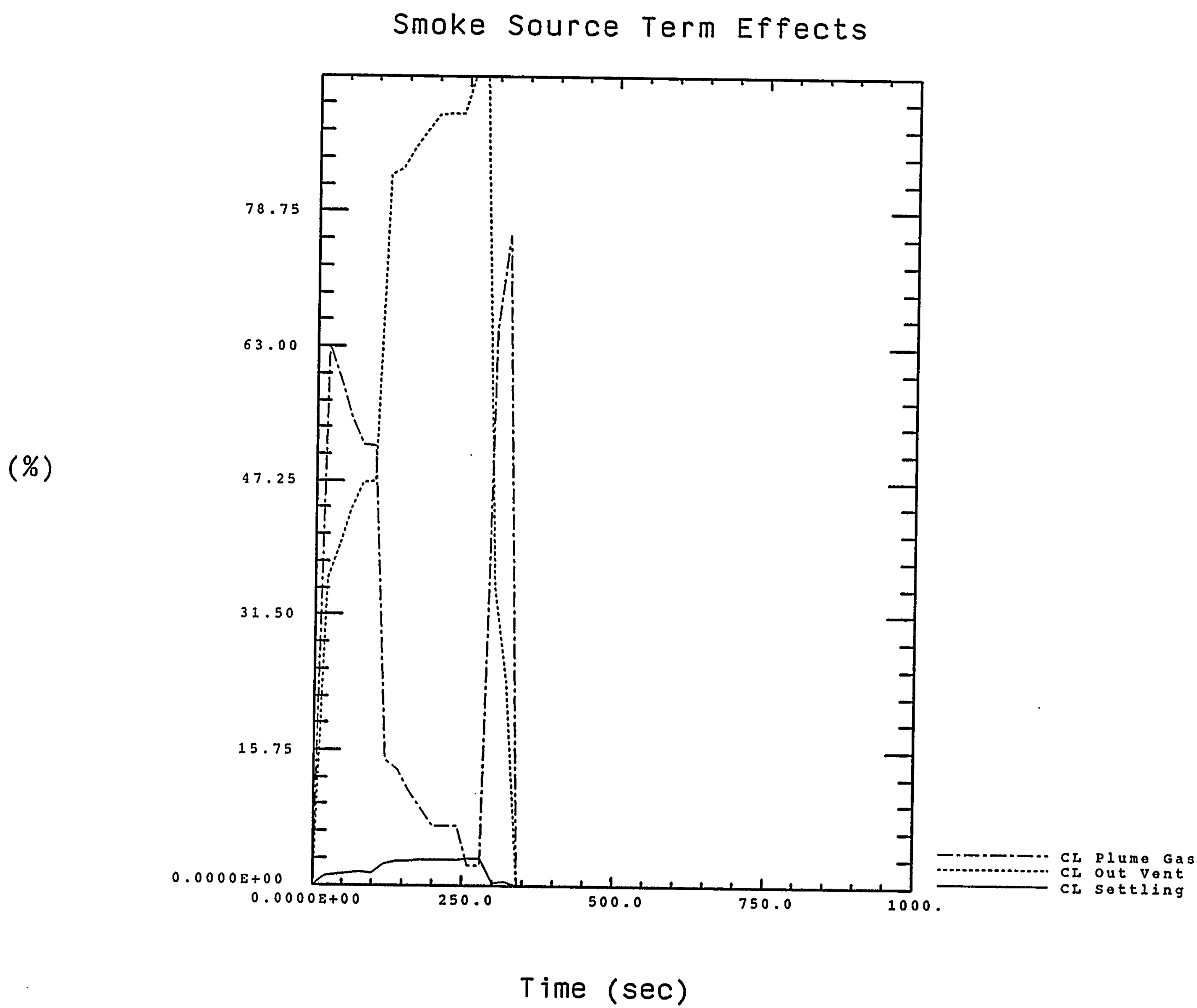



Smoke Source Term
(grams)
2.048
1.536
1.024
0.5120
0.0000E+00
Environment Accumulation
Vents Accumulation
Ceiling Accumulation
Wall Accumulation
Floor Accumulation
(%)
74.42
49.61
24.81
0.0000E+00
HL Out Vent
Diffusiophoresis
Thermophoresis
Brownian Diffusion
HL Settling
(grams)
8.762
5.842
2.921
0.0000E+00
HL Airborne t
Losses
Plume Entrainment
Fire Generated
HL Airborne t0
(%)
94.50
70.88
47.25
23.63
0.0000E+00
CL Plume Gas
CL Out Vent
CL Settling
(grams)
9.0378E-04
6.7784E-04
4.5189E-04
2.2595E-04
0.0000E+00
CL Airborne t
CL Total Losses
HL Settling
CL Airborne t0
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)


Smoke Source Term Effects
(grams)
2.048
1.707
1.365
1.024
0.6826
0.3413
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Environment Accumulation
Vents Accumulation
Ceiling Accumulation
Wall Accumulation
Floor Accumulation

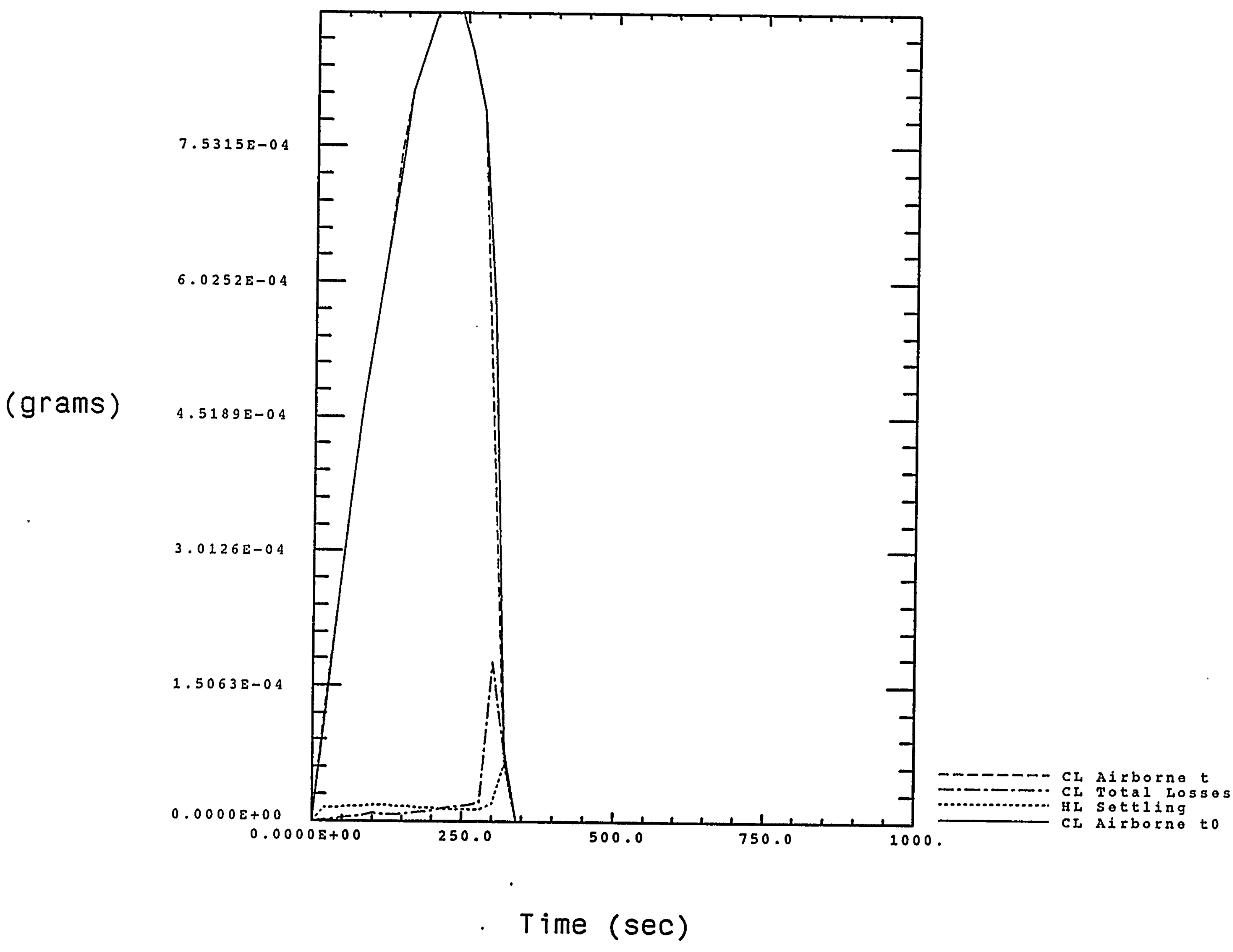

Smoke Source Term Effects
(%)
99.22
82.68
66.15
49.61
33.07
16.54
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
HL Out Vent
Diffusiophoresis
Thermophoresis
Brownian Diffusion
HL Settling
Time (sec)

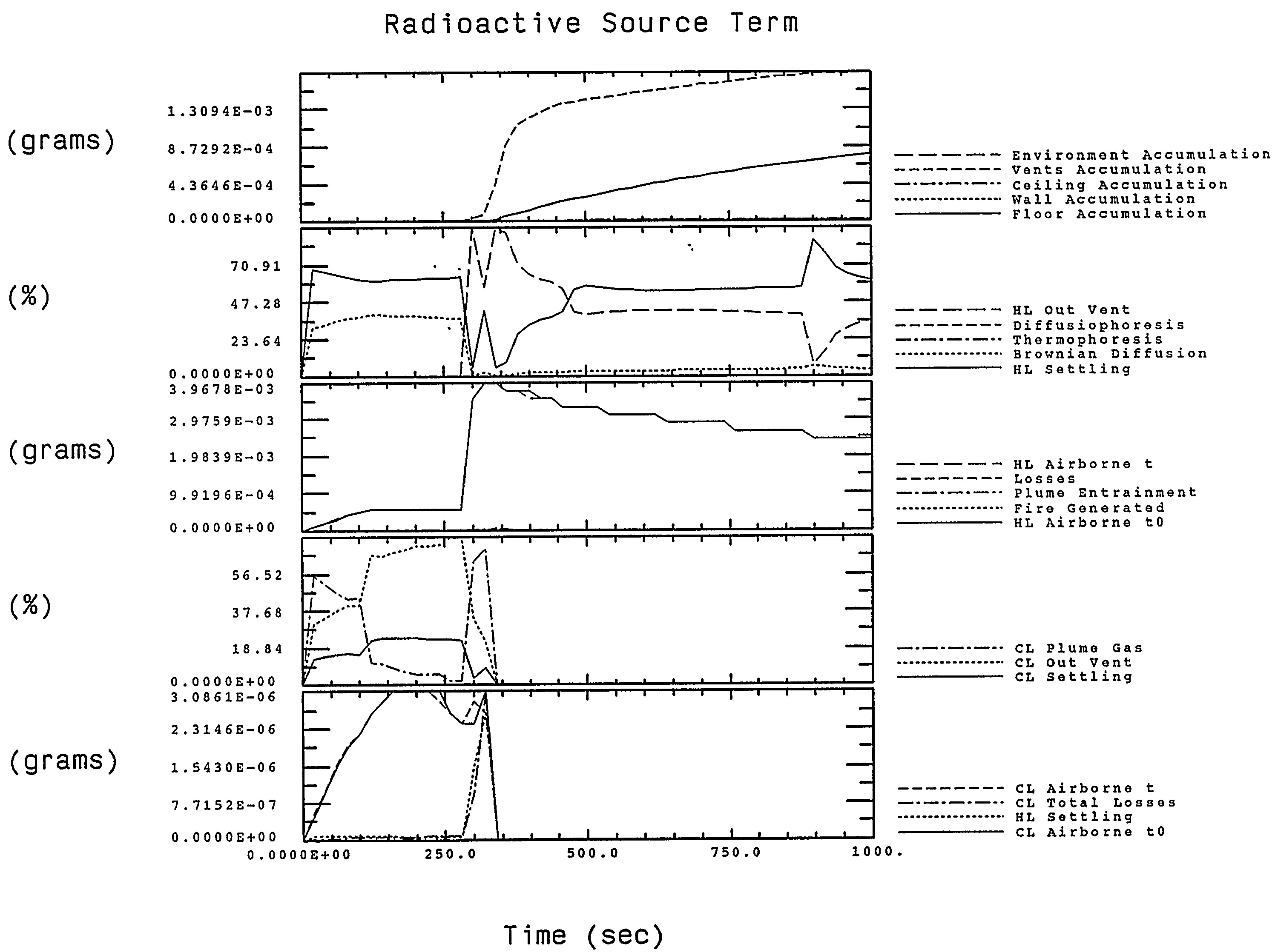

Smoke Source Term Effects
(grams)
1168.
973.6
778.9
584.2
389.4
194.7
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
HL Airborne t
Losses
Plume Entrainment
Fire Generated
HL Airborne t0



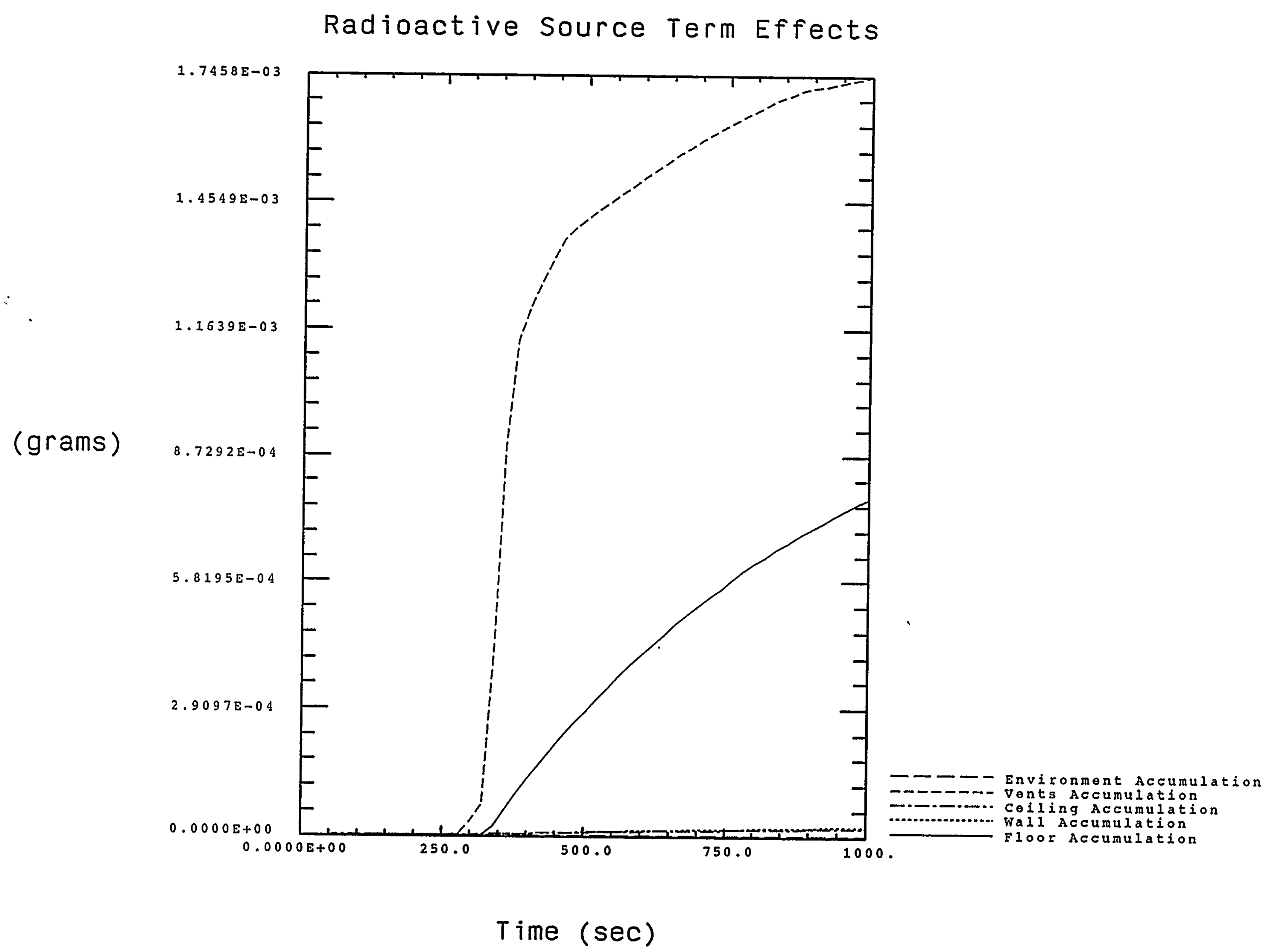
Smoke Source Term Effects
(%)
78.75
63.00
47.25
31.50
15.75
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
CL Plume Gas
CL Out Vent
CL Settling

Smoke Source Term Effects

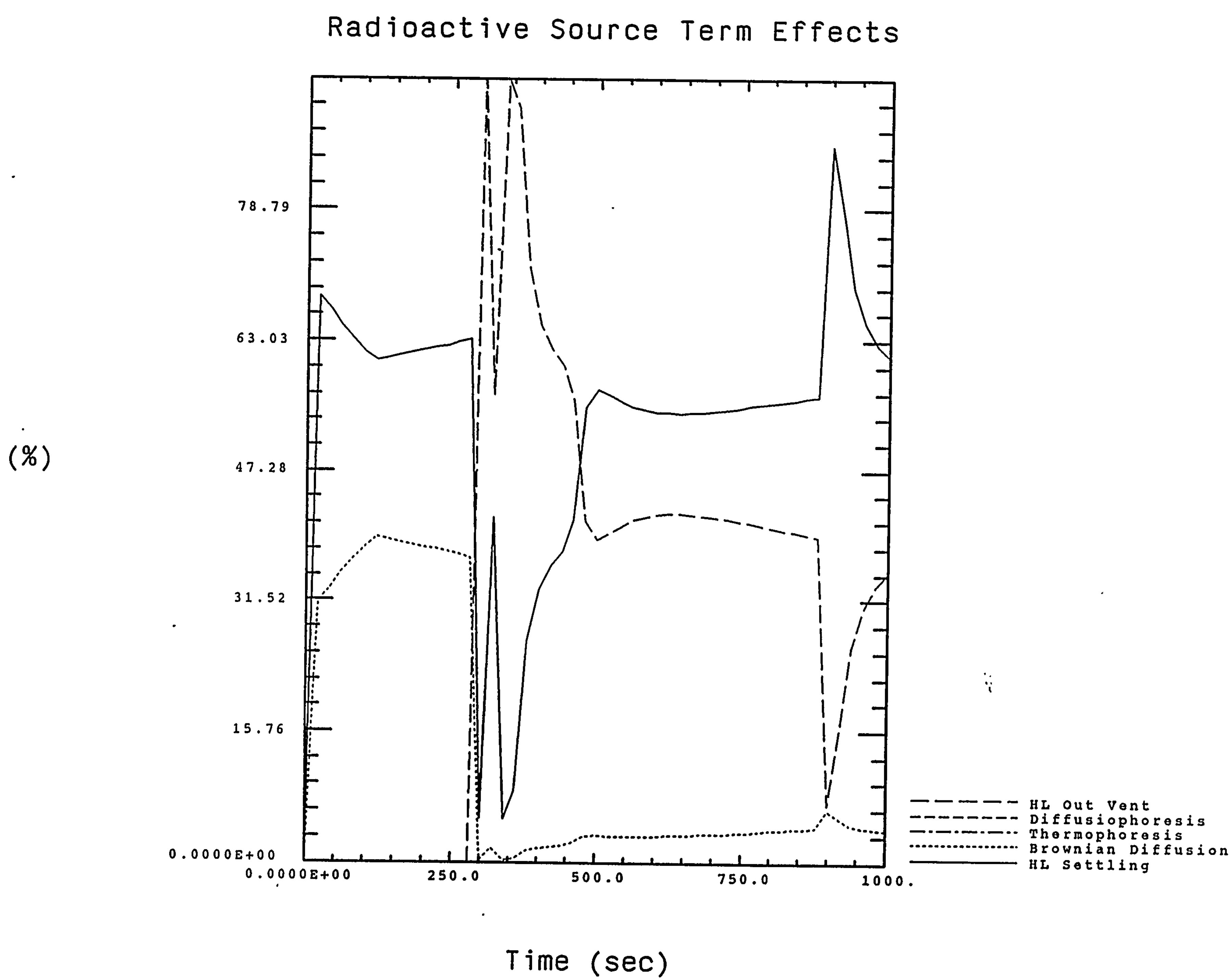




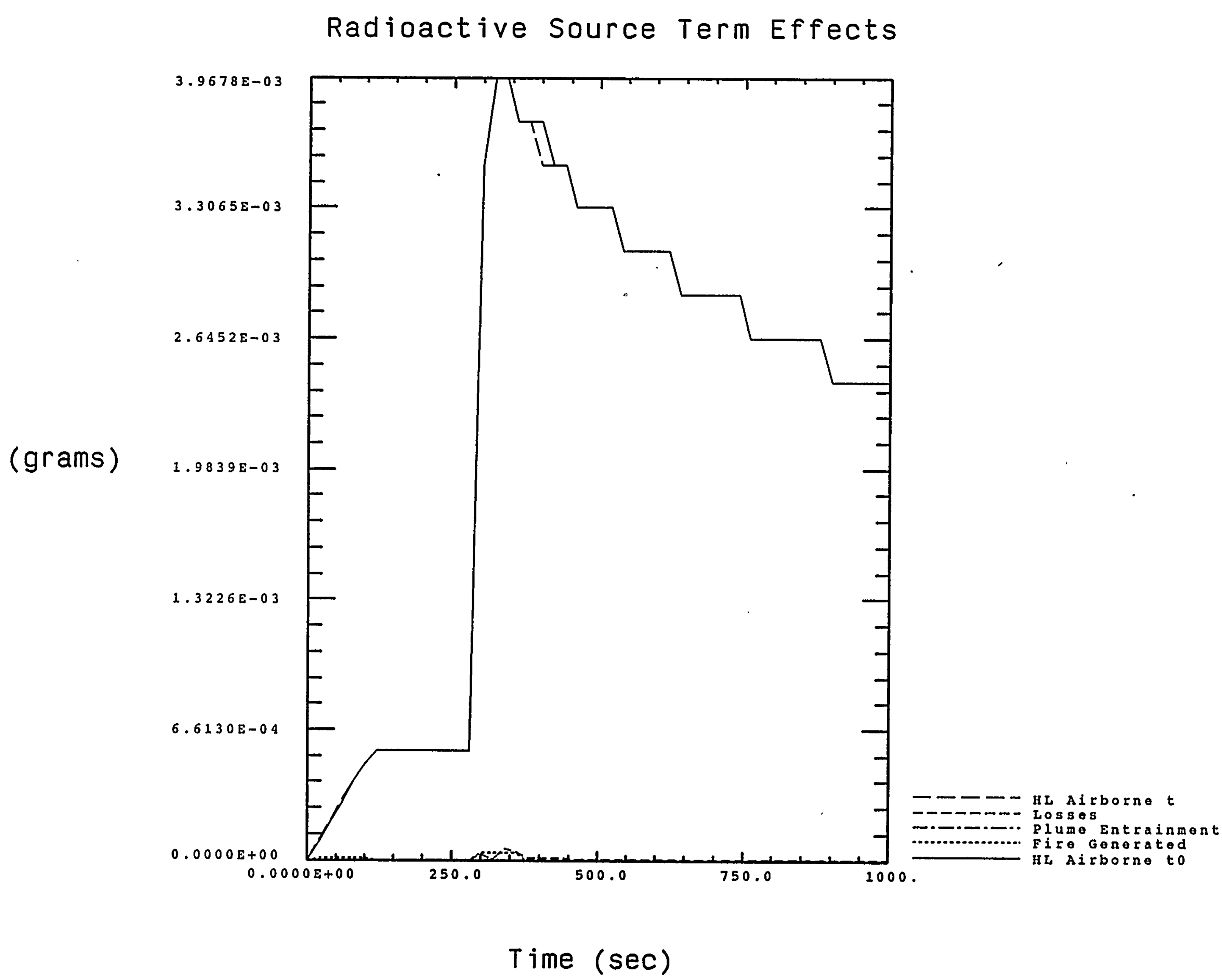
Radioactive Source Term
(grams)
1.3094E-03
8.7292E-04
4.3646E-04
0.0000E+00
Environment Accumulation
Vents Accumulation
Ceiling Accumulation
Wall Accumulation
Floor Accumulation
(%)
70.91
47.28
23.64
0.0000E+00
HL Out Vent
Diffusiophoresis
Thermophoresis
Brownian Diffusion
HL Settling
(grams)
3.9678E-03
2.9759E-03
1.9839E-03
9.9196E-04
0.0000E+00
HL Airborne t
Losses
Plume Entrainment
Fire Generated
HL Airborne t0
(%)
56.52
37.68
18.84
0.0000E+00
CL Plume Gas
CL Out Vent
CL Settling
(grams)
3.0861E-06
2.3146E-06
1.5430E-06
7.7152E-07
0.0000E+00
CL Airborne t
CL Total Losses
HL Settling
CL Airborne t0
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)



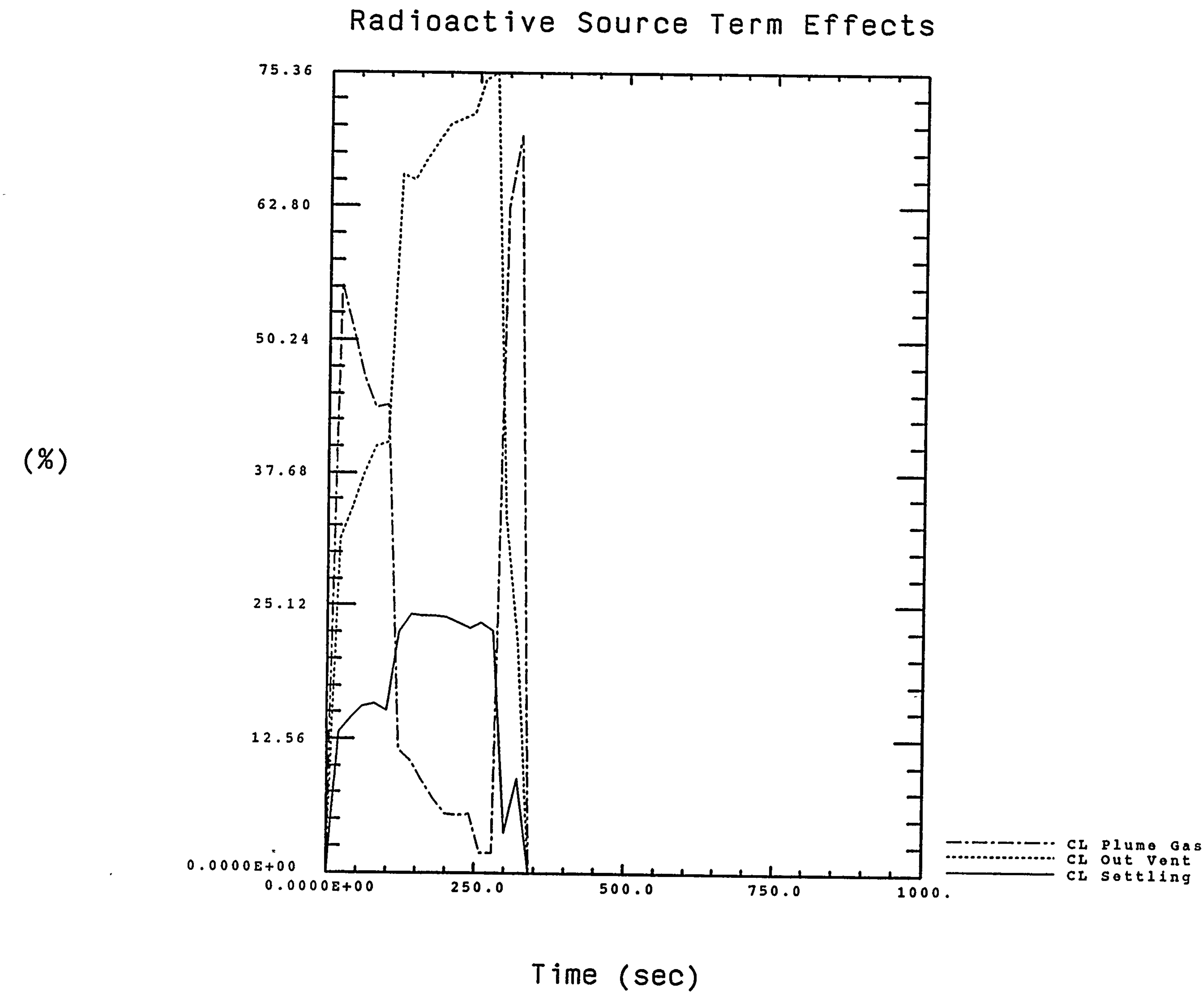
Radioactive Source Term Effects
(grams)
1.7458E-03
1.4549E-03
1.1639E-03
8.7292E-04
5.8195E-04
2.9097E-04
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Environment Accumulation
Vents Accumulation
Ceiling Accumulation
Wall Accumulation
Floor Accumulation



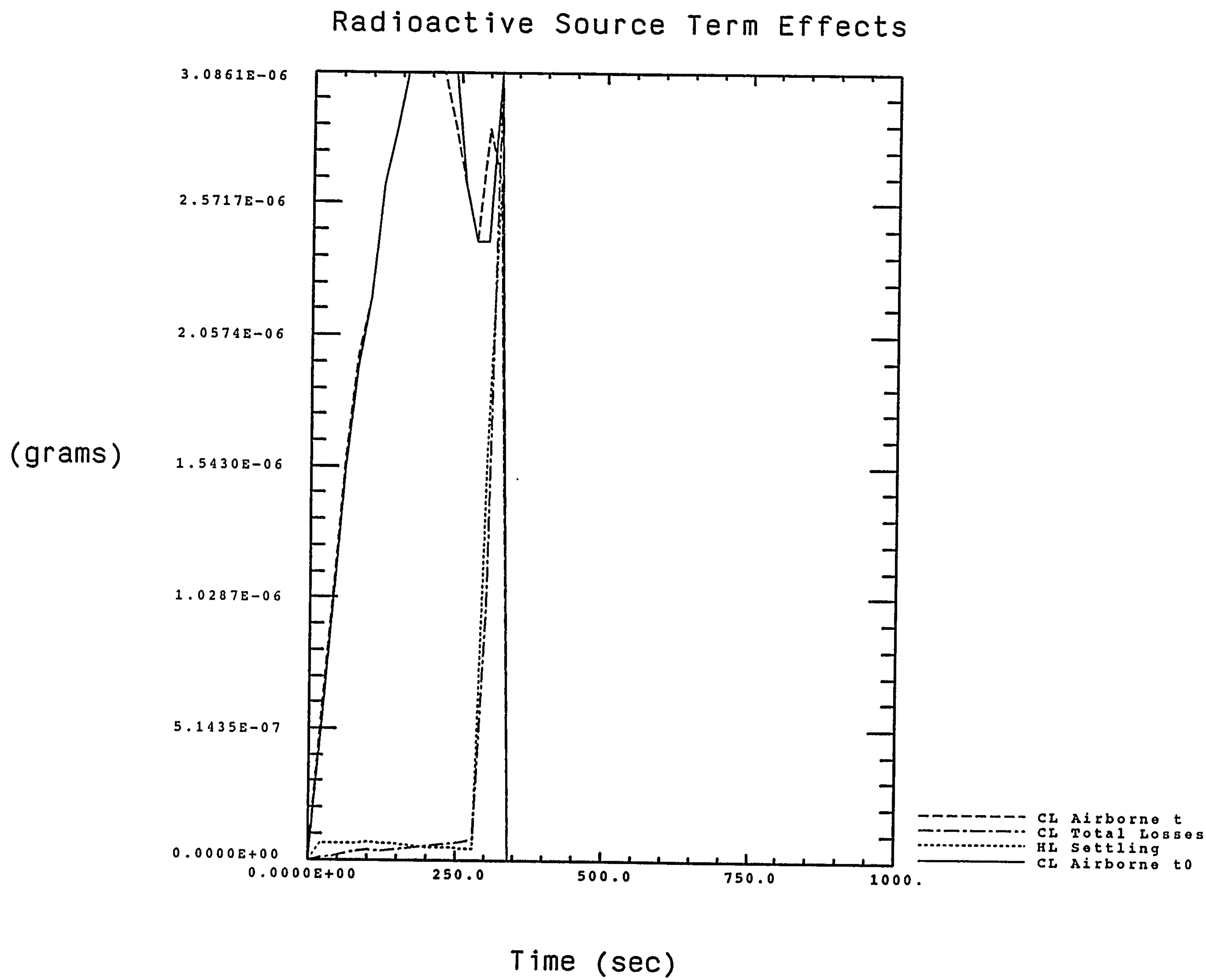
Radioactive Source Term Effects
(%)
78.79
63.03
47.28
31.52
15.76
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
HL Out Vent
Diffusiophoresis
Thermophoresis
Brownian Diffusion
HL Settling

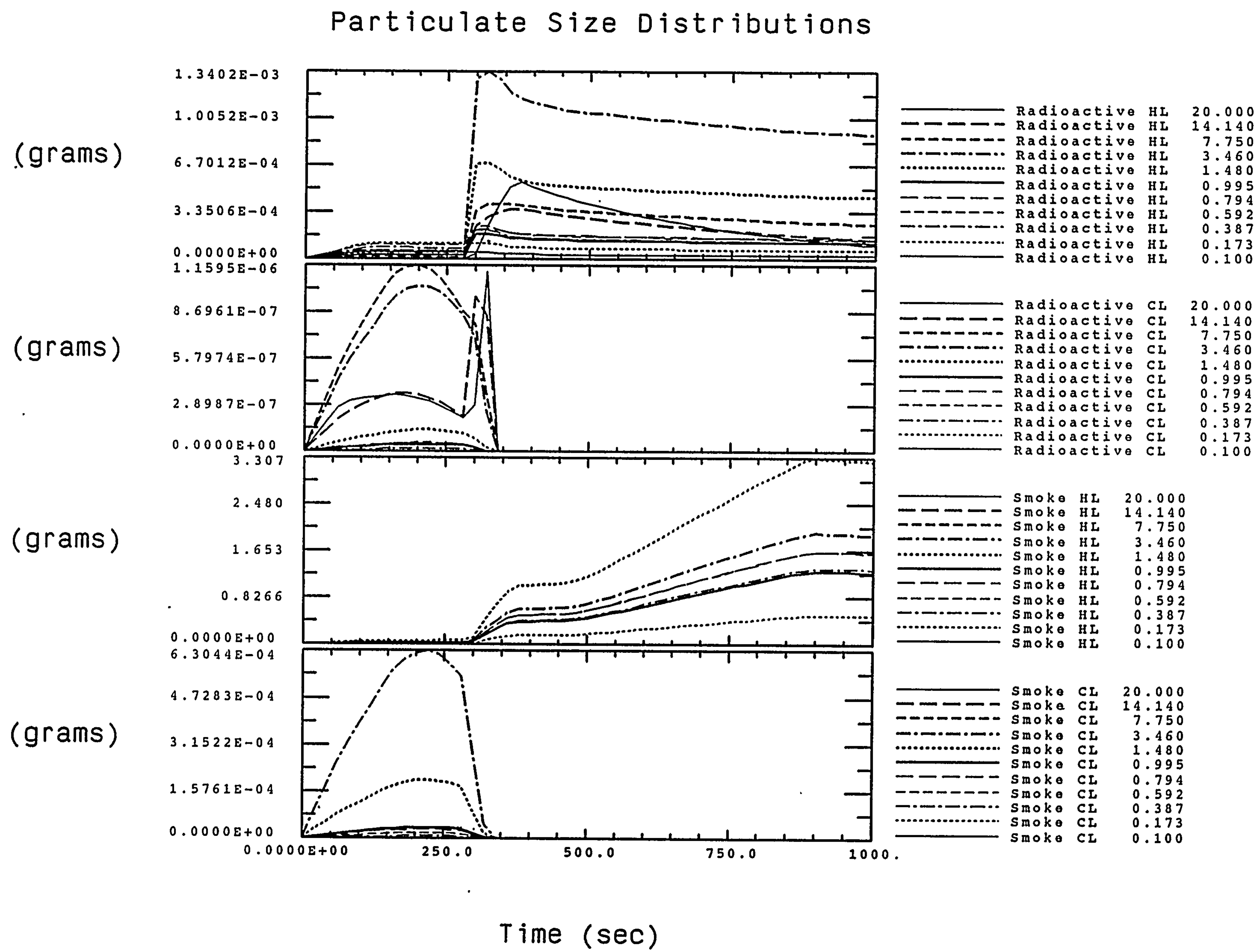


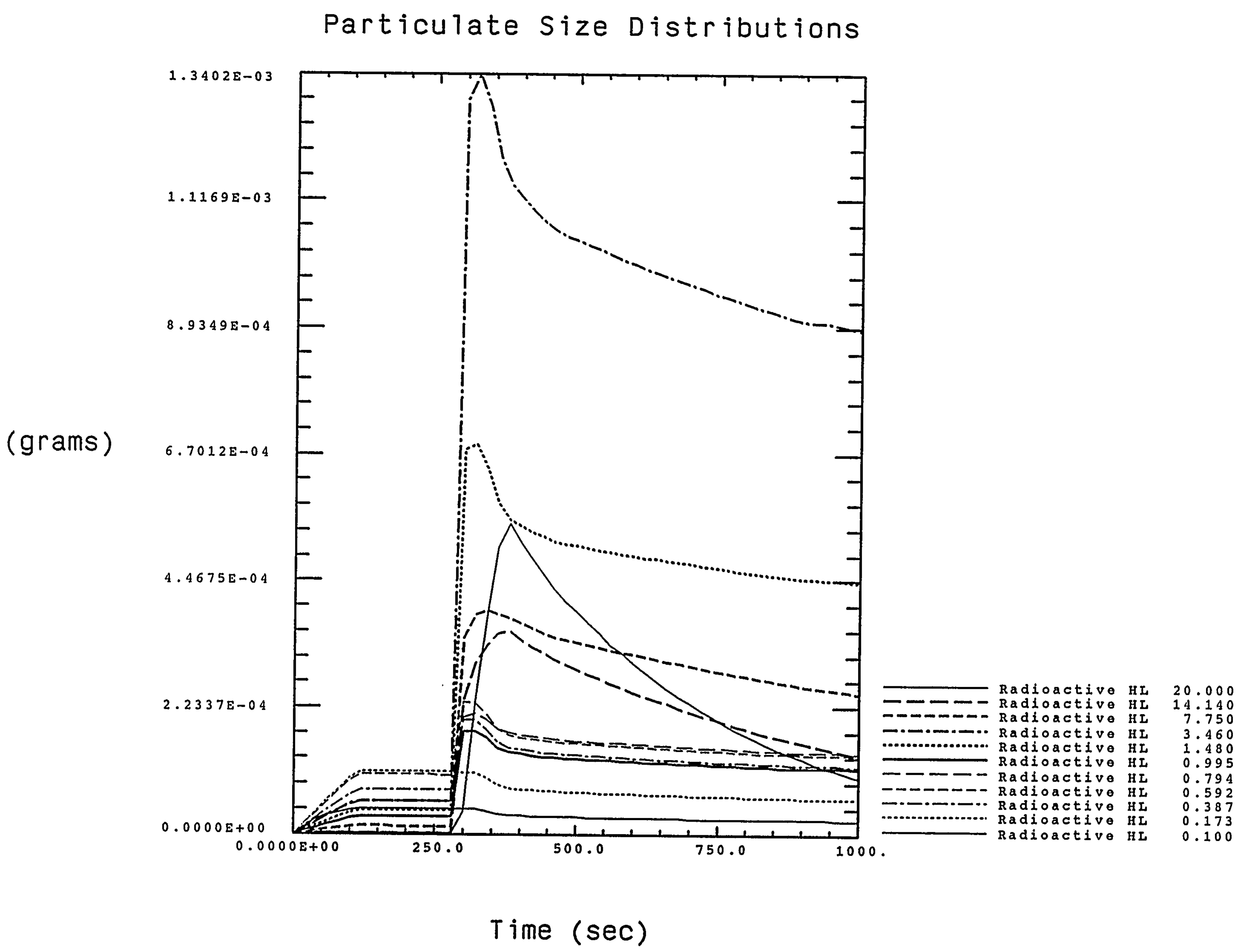
Radioactive Source Term Effects
(grams)
3.9678E-03
3.3065E-03
2.6452E-03
1.9839E-03
1.3226E-03
6.6130E-04
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
HL Airborne t
Losses
Plume Entrainment
Fire Generated
HL Airborne t0



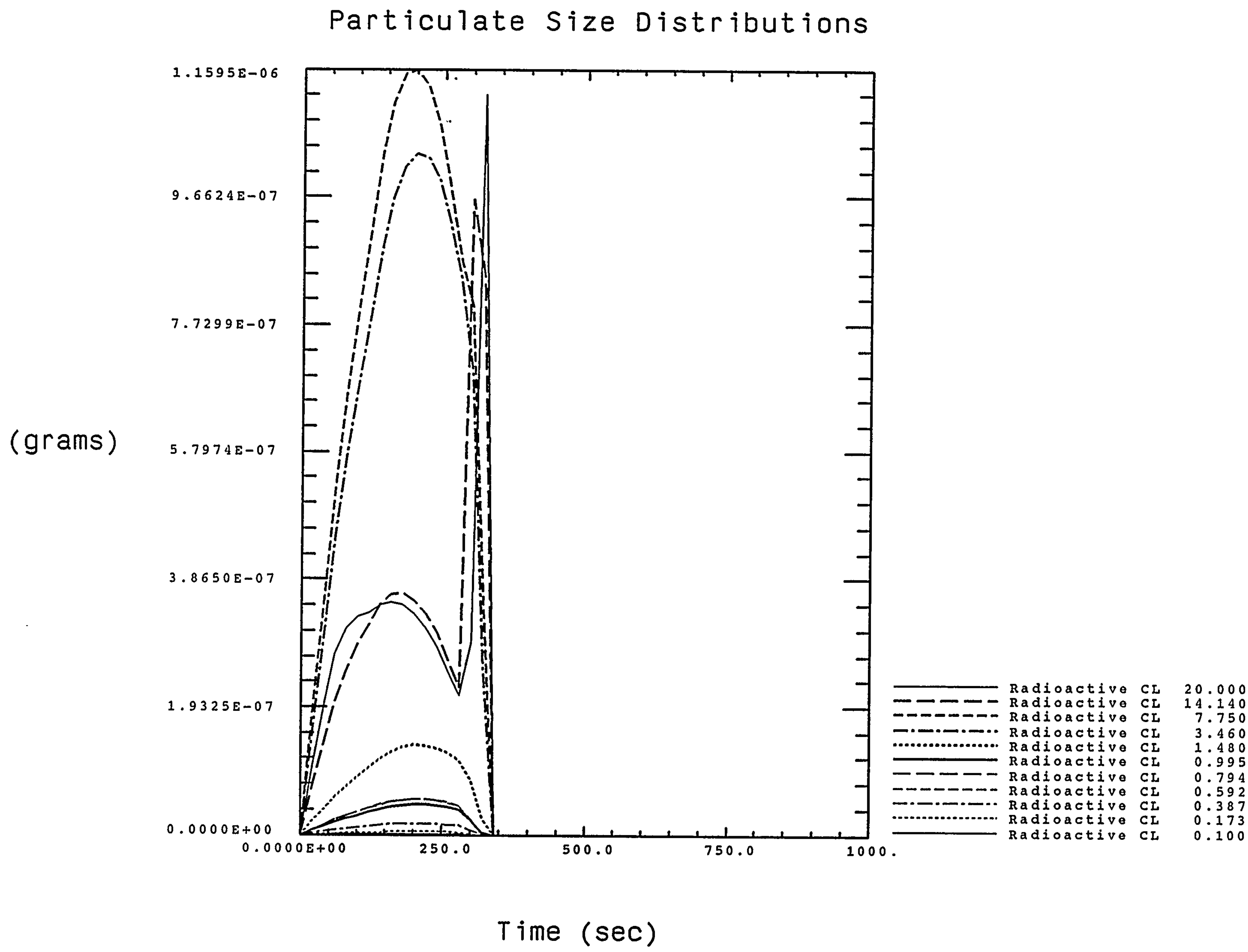
Radioactive Source Term Effects
75.36
62.80
50.24
37.68
25.12
12.56
0.0000E+00
(%)
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
CL Plume Gas
CL Out Vent
CL Settling


Radioactive Source Term Effects
(grams)
3.0861E-06
2.5717E-06
2.0574E-06
1.5430E-06
1.0287E-06
5.1435E-07
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
CL Airborne t
CL Total Losses
HL Settling
CL Airborne t0

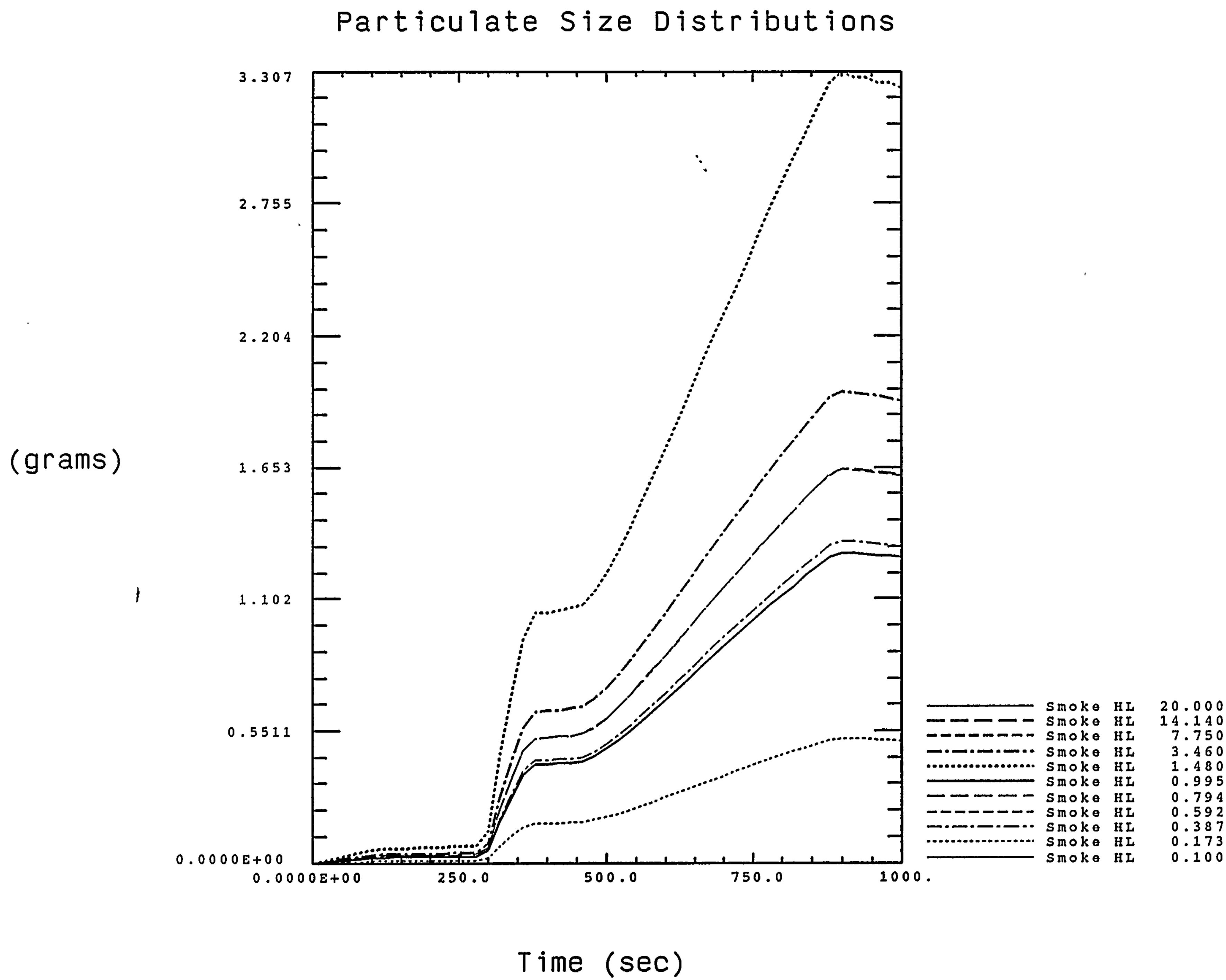

Particulate Size Distributions
(grams)
(grams)
(grams)
(grams)
1.3402E-03
1.0052E-03
6.7012E-04
3.3506E-04
0.0000E+00
1.1595E-06
8.6961E-07
5.7974E-07
2.8987E-07
0.0000E+00
3.307
2.480
1.653
0.8266
0.0000E+00
6.3044E-04
4.7283E-04
3.1522E-04
1.5761E-04
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Radioactive HL 20.000
Radioactive HL 14.140
Radioactive HL 7.750
Radioactive HL 3.460
Radioactive HL 1.480
Radioactive HL 0.995
Radioactive HL 0.794
Radioactive HL 0.592
Radioactive HL 0.387
Radioactive HL 0.173
Radioactive HL 0.100
Radioactive CL 20.000
Radioactive CL 14.140
Radioactive CL 7.750
Radioactive CL 3.460
Radioactive CL 1.480
Radioactive CL 0.995
Radioactive CL 0.794
Radioactive CL 0.592
Radioactive CL 0.387
Radioactive CL 0.173
Radioactive CL 0.100
Smoke HL 20.000
Smoke HL 14.140
Smoke HL 7.750
Smoke HL 3.460
Smoke HL 1.480
Smoke HL 0.995
Smoke HL 0.794
Smoke HL 0.592
Smoke HL 0.387
Smoke HL 0.173
Smoke HL 0.100
Smoke CL 20.000
Smoke CL 14.140
Smoke CL 7.750
Smoke CL 3.460
Smoke CL 1.480
Smoke CL 0.995
Smoke CL 0.794
Smoke CL 0.592
Smoke CL 0.387
Smoke CL 0.173
Smoke CL 0.100



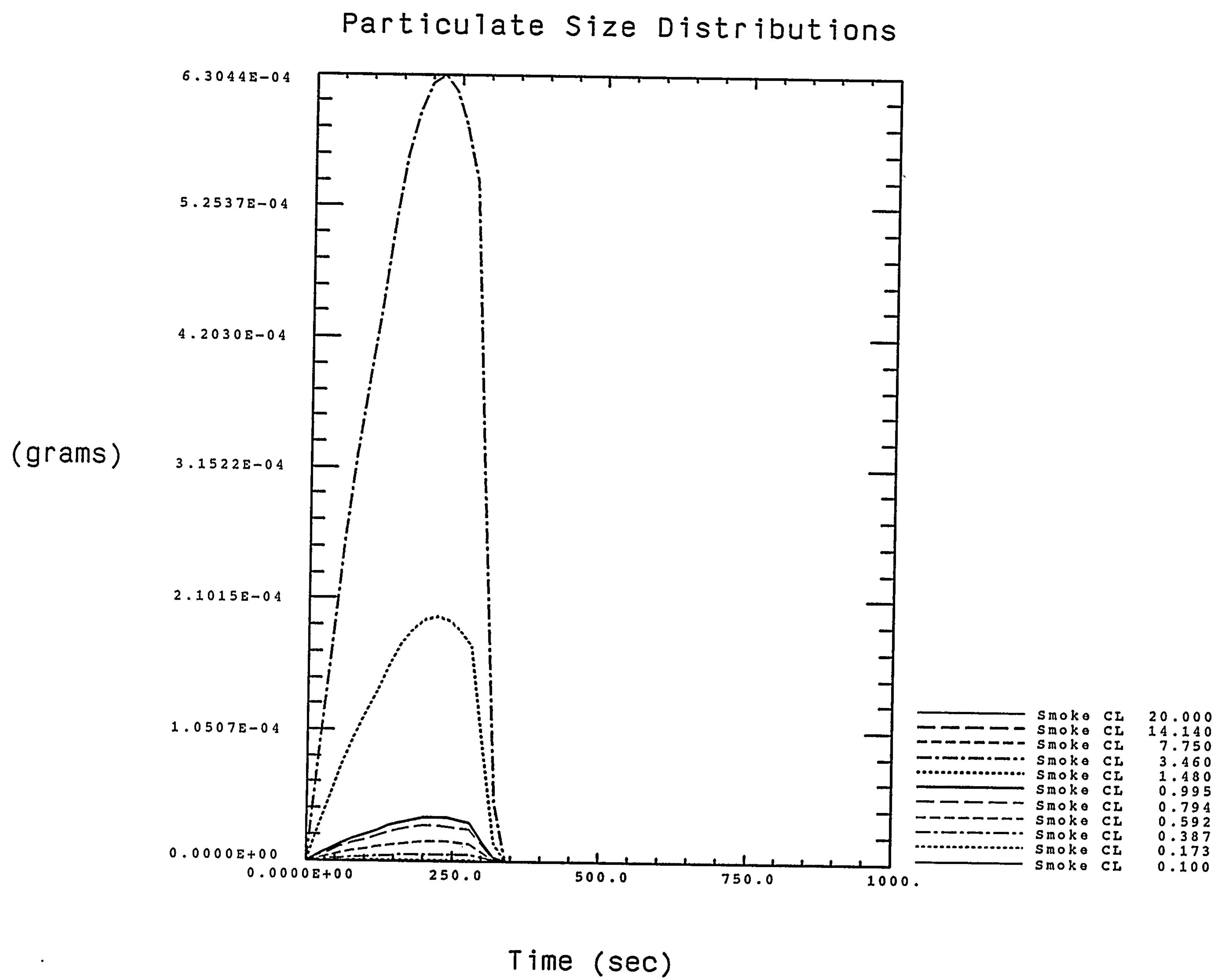
Particulate Size Distributions
(grams)
1.3402E-03
1.1169E-03
8.9349E-04
6.7012E-04
4.4675E-04
2.2337E-04
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Radioactive HL 20.000
Radioactive HL 14.140
Radioactive HL 7.750
Radioactive HL 3.460
Radioactive HL 1.480
Radioactive HL 0.995
Radioactive HL 0.794
Radioactive HL 0.592
Radioactive HL 0.387
Radioactive HL 0.173
Radioactive HL 0.100



Particulate Size Distributions
(grams)
1.1595E-06
9.6624E-07
7.7299E-07
5.7974E-07
3.8650E-07
1.9325E-07
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Radioactive CL 20.000
Radioactive CL 14.140
Radioactive CL 7.750
Radioactive CL 3.460
Radioactive CL 1.480
Radioactive CL 0.995
Radioactive CL 0.794
Radioactive CL 0.592
Radioactive CL 0.387
Radioactive CL 0.173
Radioactive CL 0.100



Particulate Size Distributions
(grams)
3.307
2.755
2.204
1.653
1.102
0.5511
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Smoke HL 20.000
Smoke HL 14.140
Smoke HL 7.750
Smoke HL 3.460
Smoke HL 1.480
Smoke HL 0.995
Smoke HL 0.794
Smoke HL 0.592
Smoke HL 0.387
Smoke HL 0.173
Smoke HL 0.100


Particulate Size Distributions
(grams)
6.3044E-04
5.2537E-04
4.2030E-04
3.1522E-04
2.1015E-04
1.0507E-04
0.0000E+00
0.0000E+00
250.0
500.0
750.0
1000.
Time (sec)
Smoke CL 20.000
Smoke CL 14.140
Smoke CL 7.750
Smoke CL 3.460
Smoke CL 1.480
Smoke CL 0.995
Smoke CL 0.794
Smoke CL 0.592
Smoke CL 0.387
Smoke CL 0.173
Smoke CL 0.100

1.5 Post Processing FIRAC Graphics

Now that we have copied the files onto the hard disk, we need to verify the installation and learn how to use the programs. Let's start with the post-processing program, POST. Enter the command

```
c:\FIRINPC> POST
```

The program will begin execution by presenting you with the main menu, from which can select either FIRIN or FIRAC graphics. Upon selecting FIRAC graphics, you will be presented with a second level menu as shown below:

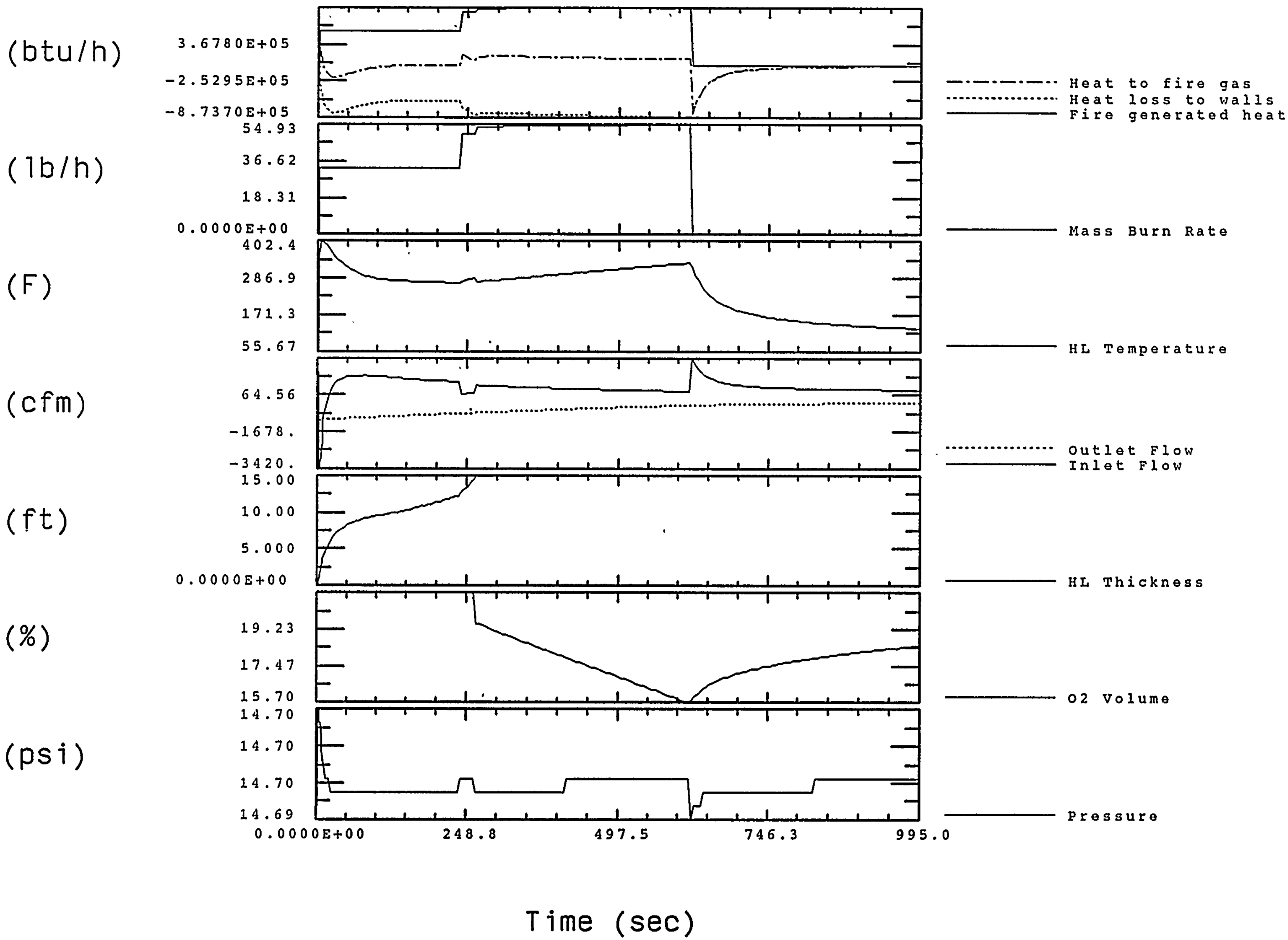
```
Which kind of graphics would you like?

 5) FIRAC Node Summary                    6) FIRAC Branch Summary
    51) Node Pressures                       61) Branch Volume flows
    52) Node Temperatures                    62) Branch Mass flows
                                             63) Branch Damper Blade Angles

99) RETURN to main menu.
```

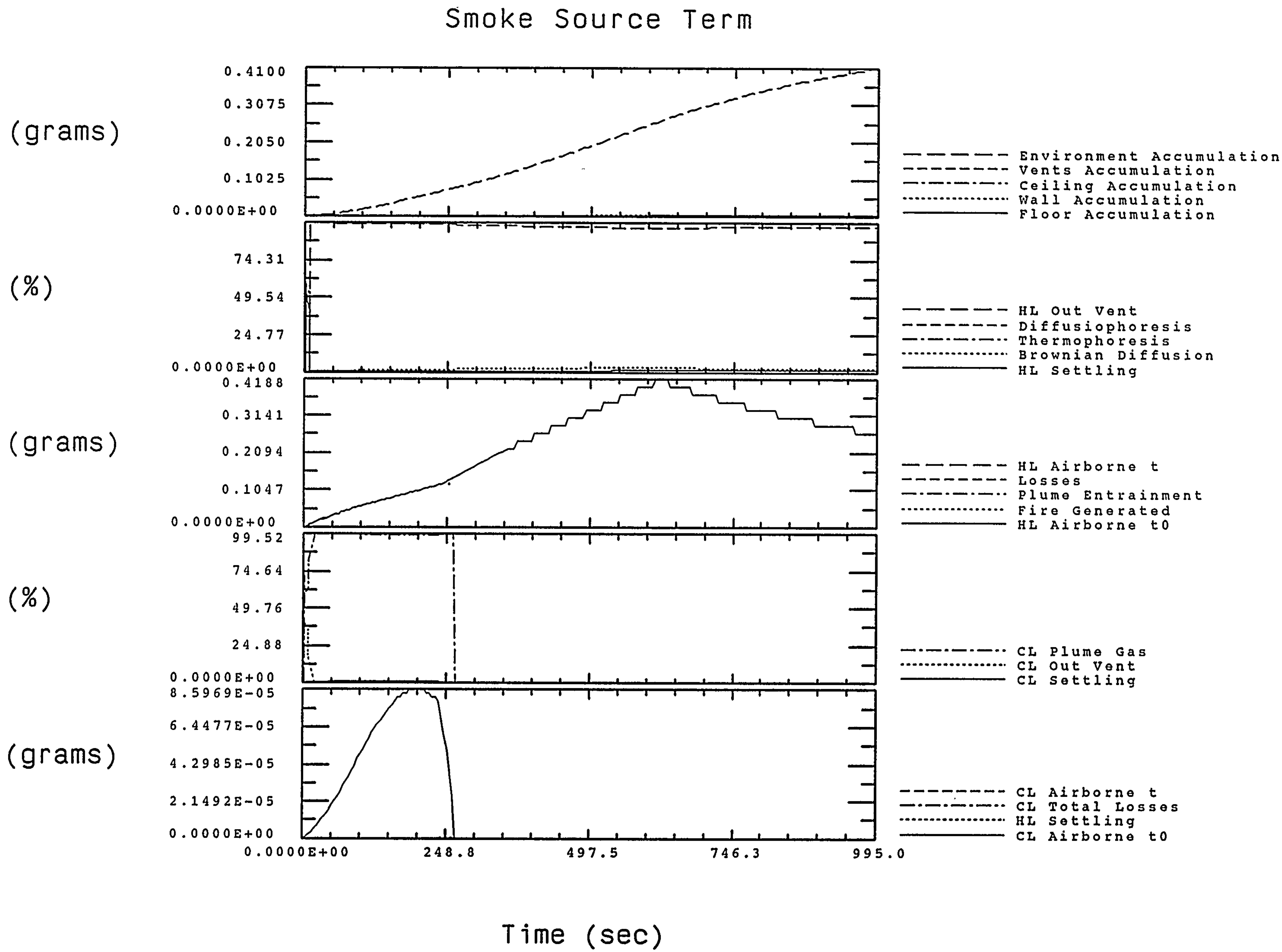
You may now choose from among the FIRAC plotting options. Let us first view plot (5), which contains a summary of the nodal response of the system. You may plot the response of a maximum of 10 nodes at a time. After entering your choice for up to 10 nodes are of interest, POST will read the output data files froma previous run of the FIRAC code, and will gather the information for presentation. At the same time, you will be able to indicate a maximum of 10 branches of interest. You may review each of these plots as desrired, and when finished, return back to the main menu. Once at the main menu, you will again have an opportunity to perform FIRIN graphics or FIRAC graphics analysis.

If you wish to review the response of other than your originally selected nodes and branches, you need to go to the main menu, and re-enter the FIRAC submenu, so that you will be prompted to enter a new set of node and branch numbers.

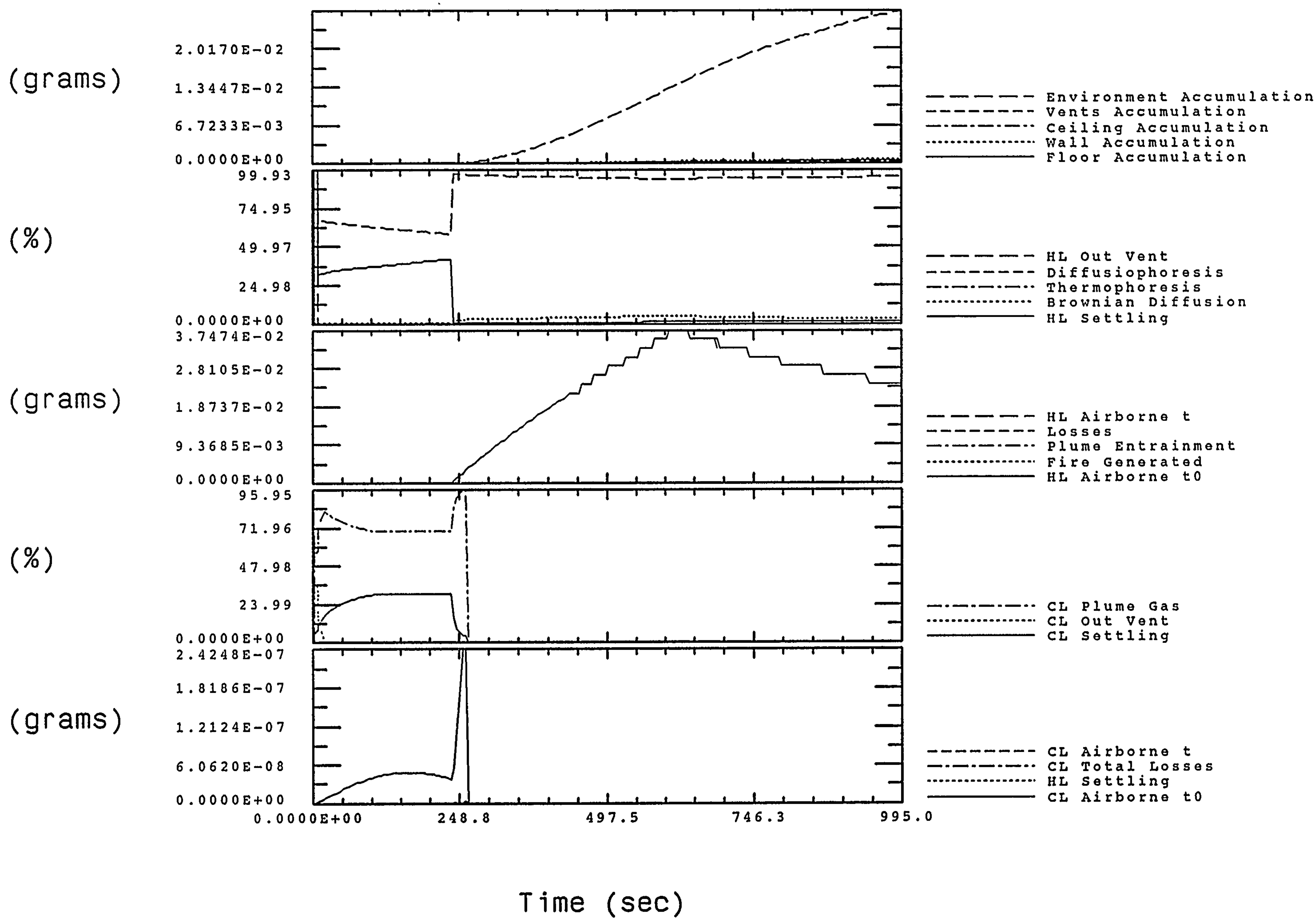
Some typical plotting results of from FIRAC sample problem number 1 are given on the following pages. Note that these results also contain information about the fire compartment response.

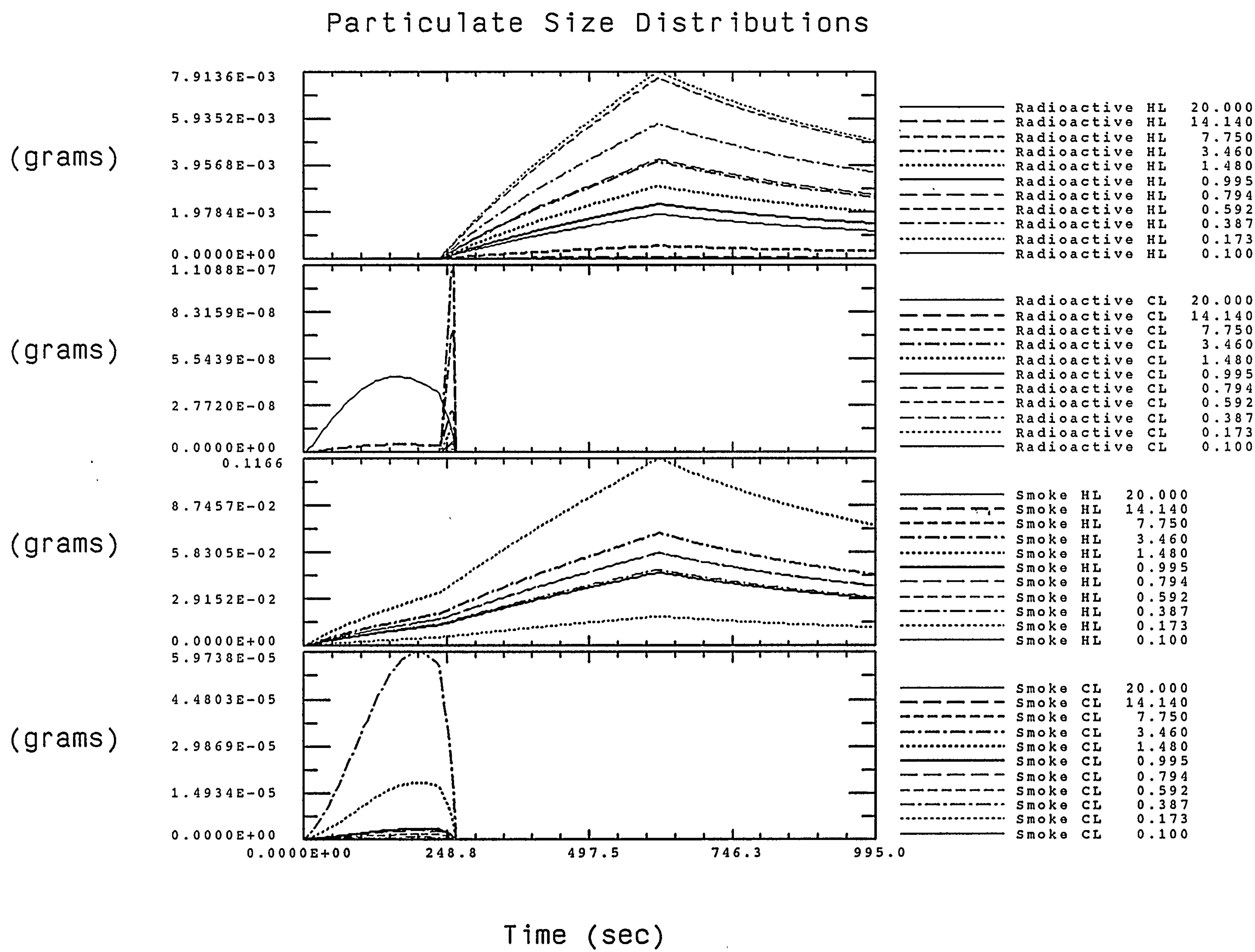
Fire Compartment Thermal Effects




Smoke Source Term
(grams)
0.4100
0.3075
0.2050
0.1025
0.0000E+00
Environment Accumulation
Vents Accumulation
Ceiling Accumulation
Wall Accumulation
Floor Accumulation
(%)
74.31
49.54
24.77
0.0000E+00
HL Out Vent
Diffusiophoresis
Thermophoresis
Brownian Diffusion
HL Settling
(grams)
0.4188
0.3141
0.2094
0.1047
0.0000E+00
HL Airborne t
Losses
Plume Entrainment
Fire Generated
HL Airborne t0
(%)
99.52
74.64
49.76
24.88
0.0000E+00
CL Plume Gas
CL Out Vent
CL Settling
(grams)
8.5969E-05
6.4477E-05
4.2985E-05
2.1492E-05
0.0000E+00
CL Airborne t
CL Total Losses
HL Settling
CL Airborne t0
0.0000E+00
248.8
497.5
746.3
995.0
Time (sec)

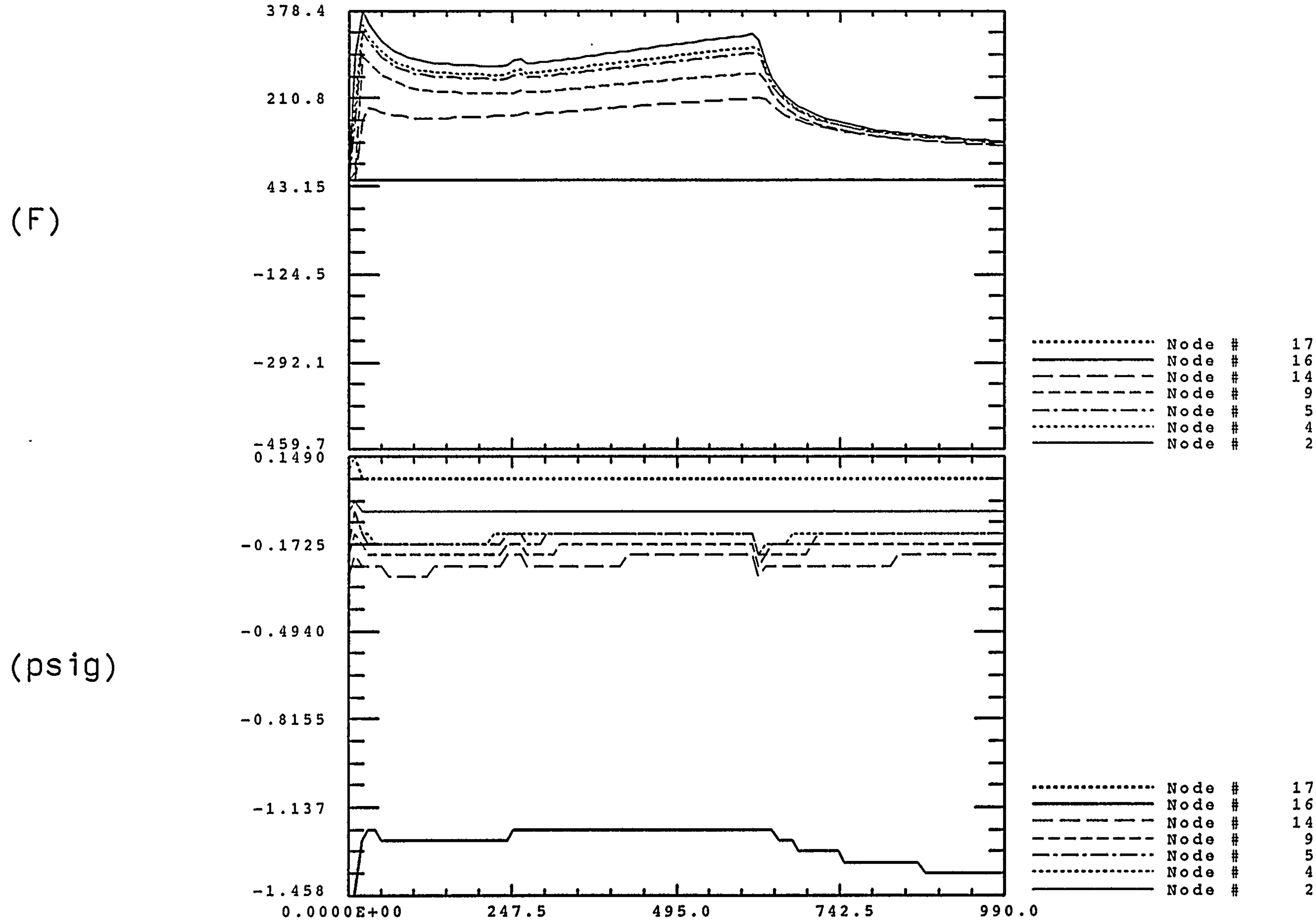
Radioactive Source Term



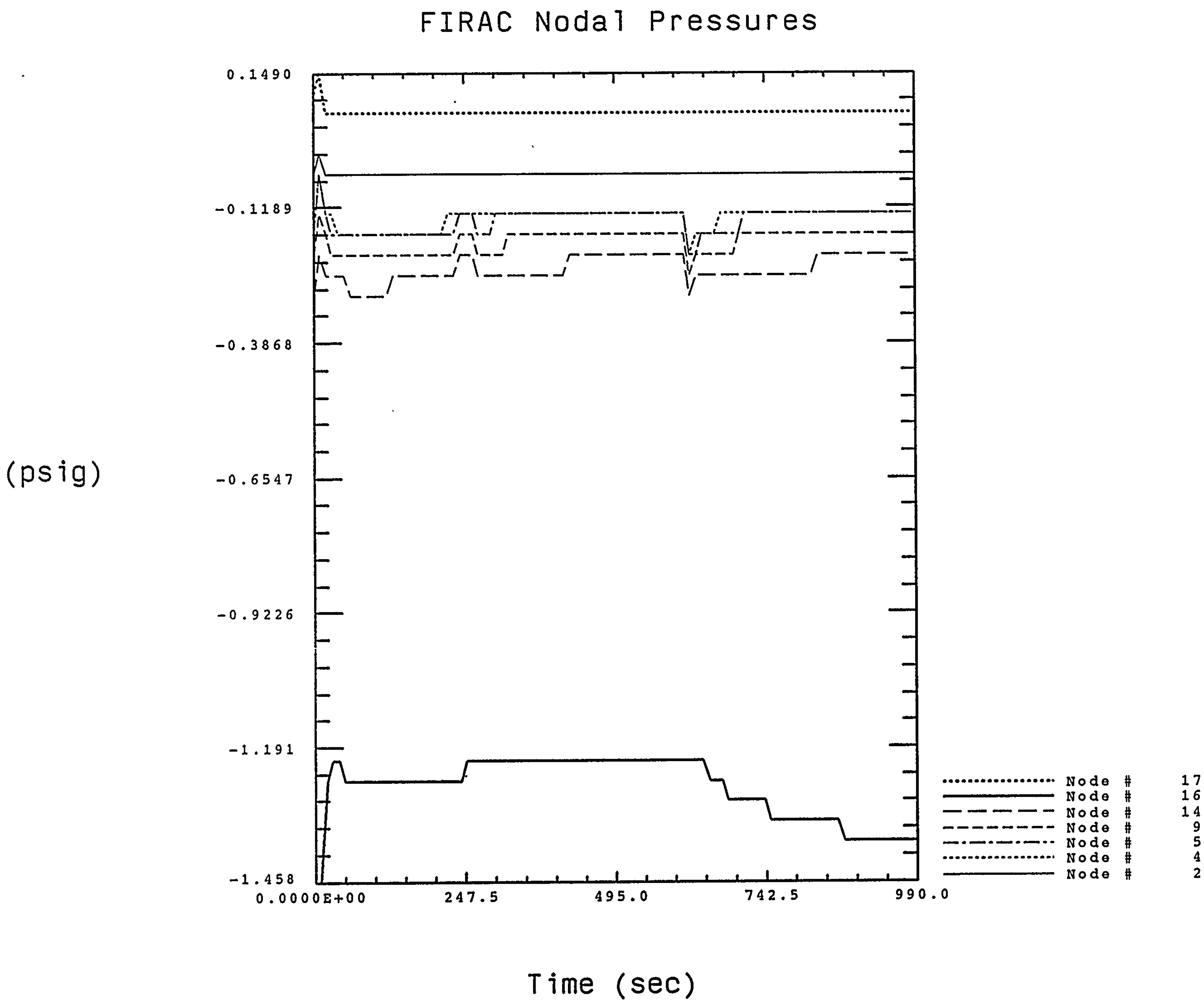


Particulate Size Distributions
(grams)
7.9136E-03
5.9352E-03
3.9568E-03
1.9784E-03
0.0000E+00
1.1088E-07
8.3159E-08
5.5439E-08
2.7720E-08
0.0000E+00
0.1166
8.7457E-02
5.8305E-02
2.9152E-02
0.0000E+00
5.9738E-05
4.4803E-05
2.9869E-05
1.4934E-05
0.0000E+00
0.0000E+00
248.8
497.5
746.3
995.0
Time (sec)
Radioactive HL 20.000
Radioactive HL 14.140
Radioactive HL 7.750
Radioactive HL 3.460
Radioactive HL 1.480
Radioactive HL 0.995
Radioactive HL 0.794
Radioactive HL 0.592
Radioactive HL 0.387
Radioactive HL 0.173
Radioactive HL 0.100
Radioactive CL 20.000
Radioactive CL 14.140
Radioactive CL 7.750
Radioactive CL 3.460
Radioactive CL 1.480
Radioactive CL 0.995
Radioactive CL 0.794
Radioactive CL 0.592
Radioactive CL 0.387
Radioactive CL 0.173
Radioactive CL 0.100
Smoke HL 20.000
Smoke HL 14.140
Smoke HL 7.750
Smoke HL 3.460
Smoke HL 1.480
Smoke HL 0.995
Smoke HL 0.794
Smoke HL 0.592
Smoke HL 0.387
Smoke HL 0.173
Smoke HL 0.100
Smoke CL 20.000
Smoke CL 14.140
Smoke CL 7.750
Smoke CL 3.460
Smoke CL 1.480
Smoke CL 0.995
Smoke CL 0.794
Smoke CL 0.592
Smoke CL 0.387
Smoke CL 0.173
Smoke CL 0.100

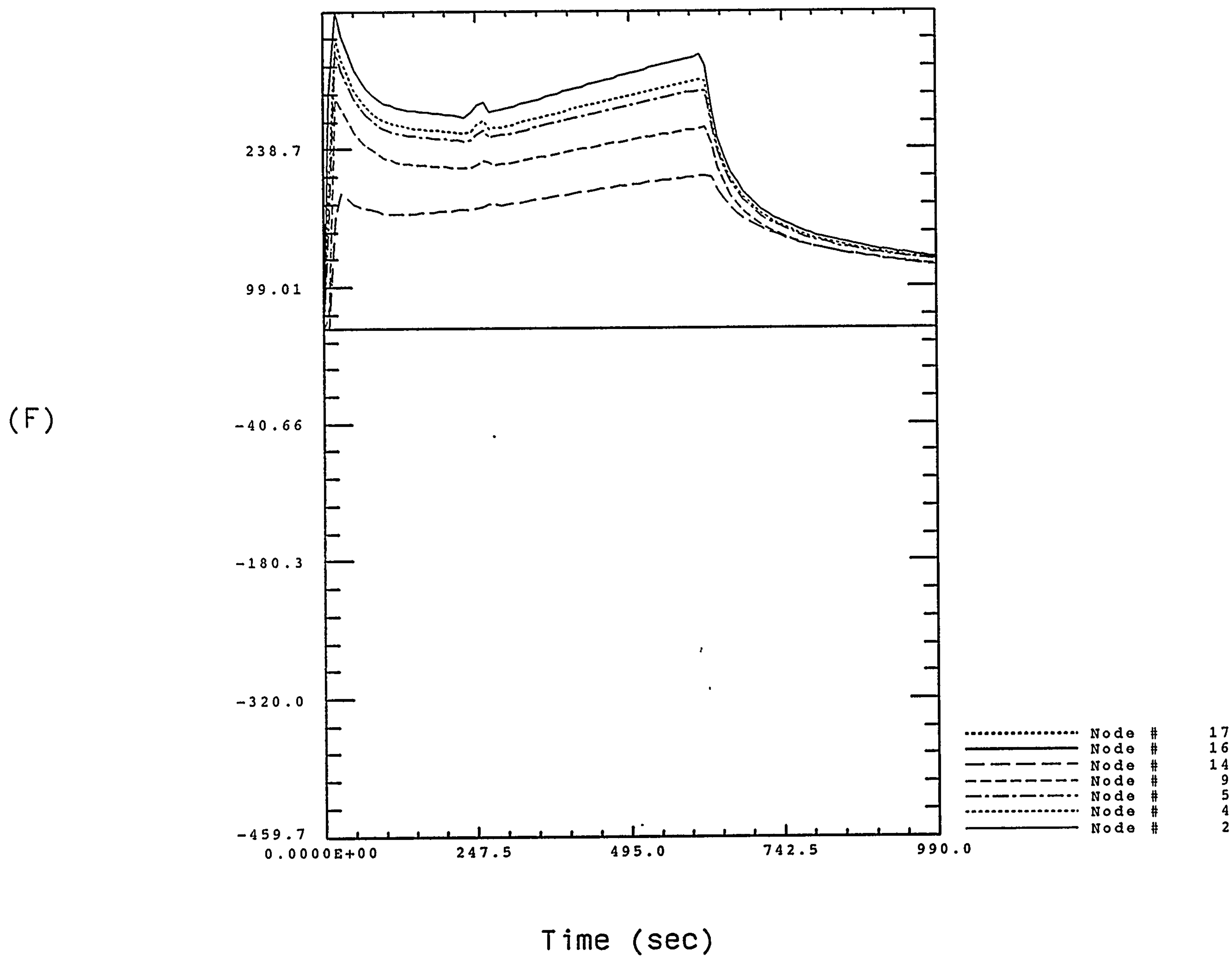
FIRAC Nodal Summary



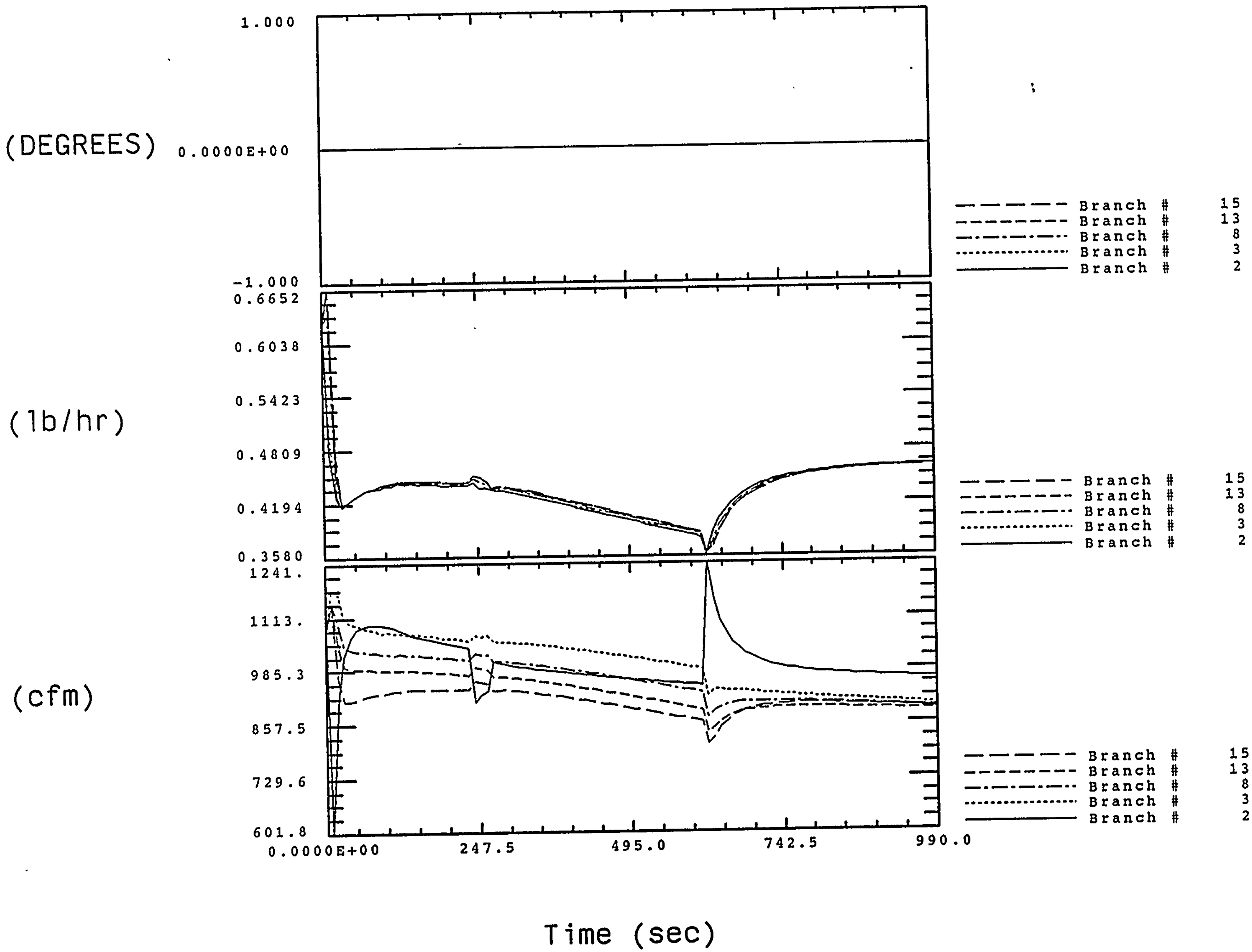
Time (sec)

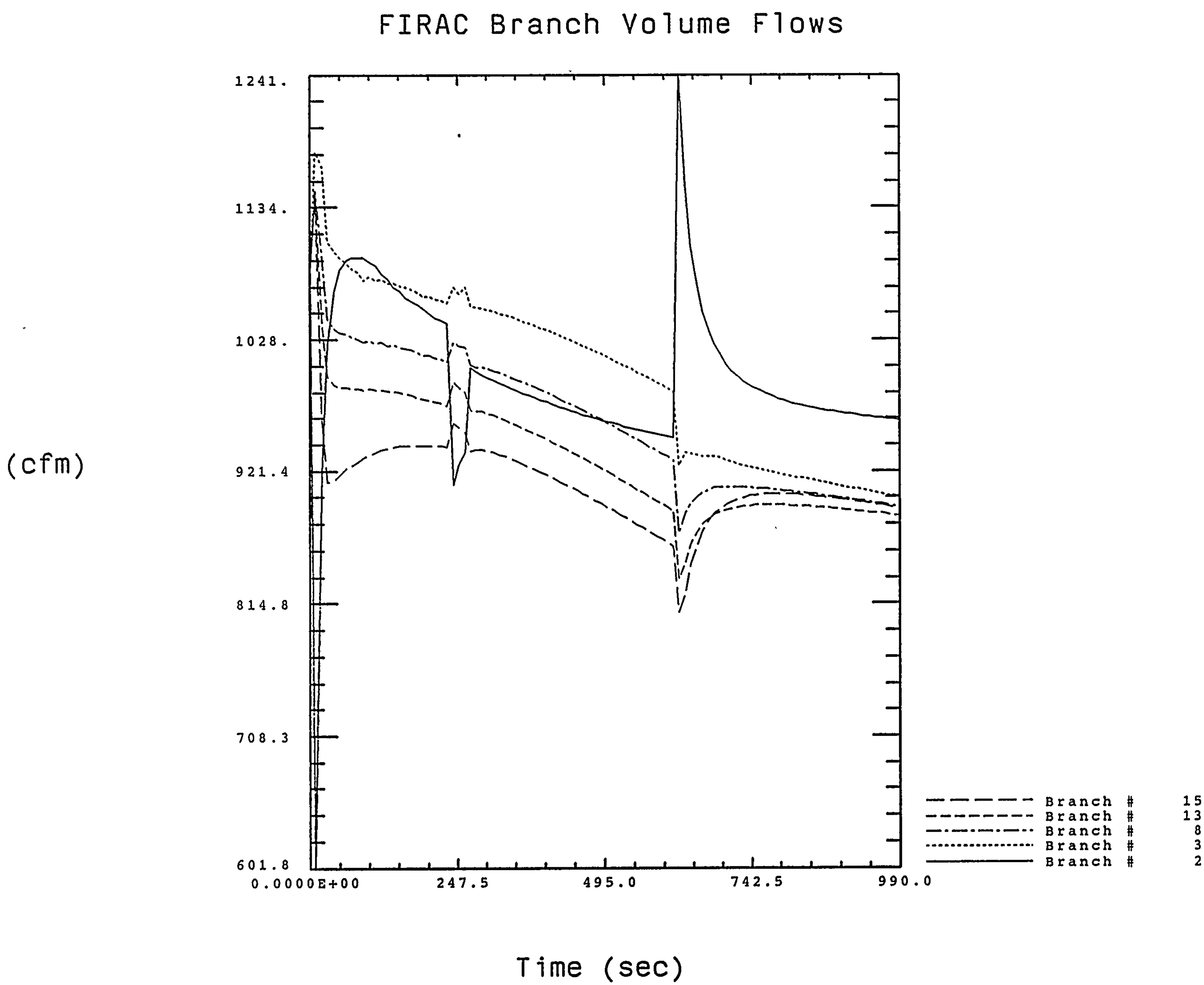

FIRAC Nodal Pressures
(psig)
0.1490
-0.1189
-0.3868
-0.6547
-0.9226
-1.191
-1.458
0.0000E+00
247.5
495.0
742.5
990.0
Time (sec)
Node # 17
Node # 16
Node # 14
Node # 9
Node # 5
Node # 4
Node # 2

FIRAC Nodal Temperatures

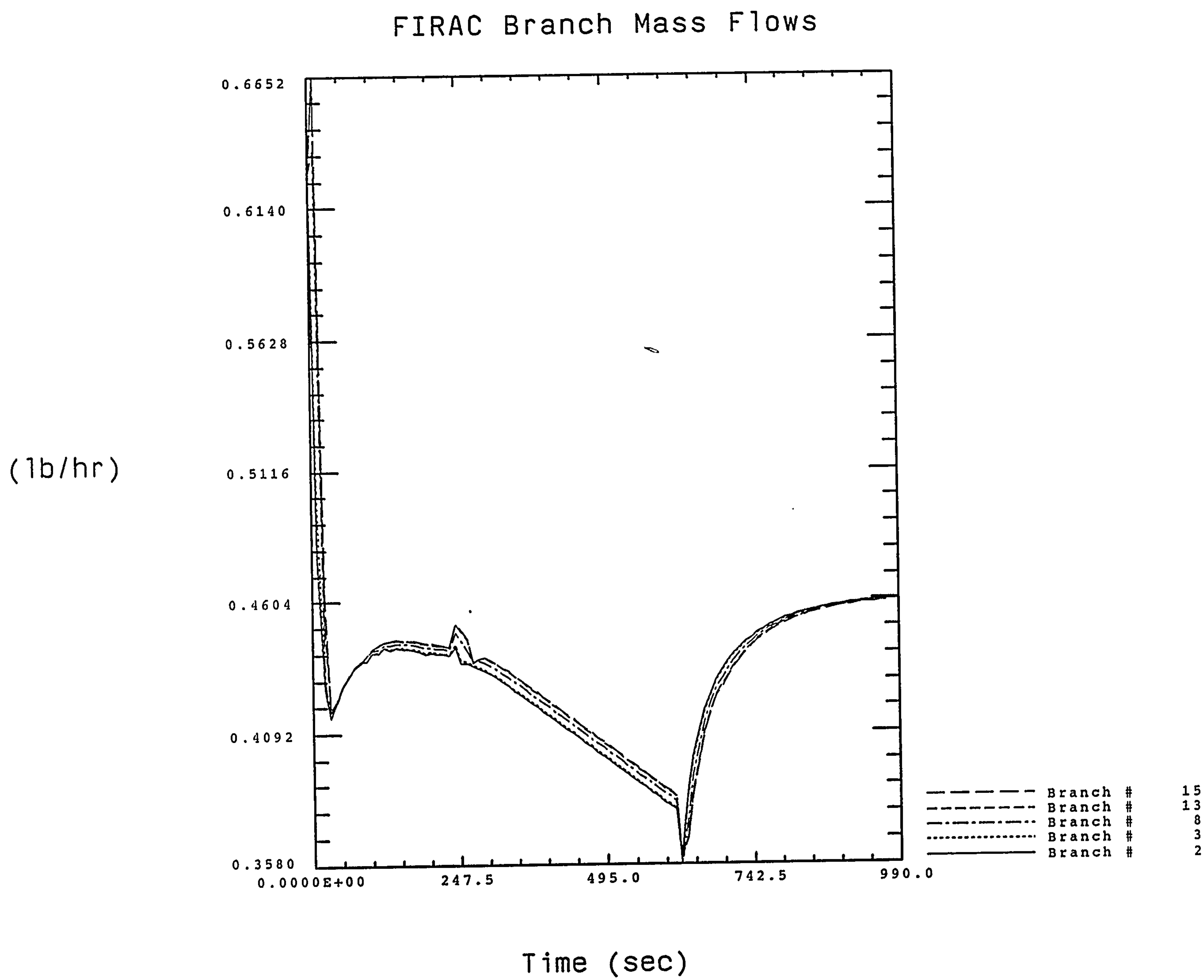


FIRAC Branch Summary





FIRAC Branch Volume Flows
(cfm)
1241.
1134.
1028.
921.4
814.8
708.3
601.8
0.0000E+00
247.5
495.0
742.5
990.0
Branch # 15
Branch # 13
Branch # 8
Branch # 3
Branch # 2
Time (sec)



FIRAC Branch Mass Flows
(lb/hr)
0.6652
0.6140
0.5628
0.5116
0.4604
0.4092
0.3580
0.0000E+00
247.5
495.0
742.5
990.0
Time (sec)
Branch # 15
Branch # 13
Branch # 8
Branch # 3
Branch # 2

1.6 Interactive FIRIN Graphics

Next we will verify the installation of the `FIRINPC` code by running it on sample problem number 1 from the FIRIN manual. Note that this guide will not serve as a substitute for the technical information contained within the FIRIN manual – this guide simply illustrates the mechanism for running the code. To begin, enter the command "`FIRINPC`" at the DOS prompt. The program will prompt you to enter the name of the input data file. You may respond by entering "`FIRE1.DAT`". Since the plotting option "PLOTANS" was declared as affirmative by a 'Y' in the first line of the input data file, `FIRINPC` will print the plotting option menu like we saw in the post processing section. Choose option "1" and press the return key. `FIRINPC` will now clear the graphics display screen, and draw seven sets of axes on the screen using a conservative set of values for the vertical axis scales in each case. As `FIRINPC` continues execution, the results will be displayed graphically on the screen.

Depending upon the speed of your machine, you may have to wait quite a while for the simulation to complete. On an 80286 based machine, your wait may be as long as 3 hours for problem 1, even with a math coprocessor. On an 80386 class machine with a coprocessor, the simulation should be completed in about 20 minutes. Your own time may be somewhat greater or less, depending upon the clock speed and disk access time for your particular machine.

After the simulation has completed, and the plots completely extend over the length of the horizontal axis, you may continue to view the summary plot as long as you wish. When finished looking at the summary plot, you may press the enter key to return to the main plotting option menu. The menu that we saw earlier will be re-displayed, and you may choose from the 25 plotting options, or enter "99" to exit the program. Feel free to review several of the plots before exiting the program. And remember, if you want to come back later and review some of the results, you can always activate the post processing program.

Next, you may wish to change the "Y" in the first line of `FIRE1.DAT` to a "N" to disable the graphics option. Then if you re-run the problem by executing `FIRINPC`, the plotting option will not be displayed. However, so that you can indeed monitor the progress of the simulation, a simple time step counter will update the screen periodically. This option is useful for running `FIRINPC` in a window on an 80386 machine. In this way, you can run `FIRINPC` in the background, and continue working on other projects in the foreground. Or, if the simulation is quite large, you may leave the machine running over night and return the next morning. After the simulation has been completed, you can use the `POST` program to review the results.

Chapter 2

Programmer's Reference

2.1 Introduction

This section is not intended to be a thorough review of the theory and development of the FIRIN code. Rather, this section will contain a summary of the files used by the `FIRINPC` code and the post processor `POST`.

2.2 Description of Files

`FGRAPH.FD` Microsoft Fortran Version 5.x include file for user programs using the graphics libraries.

`FGRAPH.FI` Microsoft Fortran Version 5.x include file for user programs using the graphics libraries.

`GRAPHICS.LIB` Microsoft Fortran Version 5.x graphics library.

`DOSCALLS.LIB` Microsoft Fortran Version 5.x DOS interface library.

`LLIBFORE.LIB` Microsoft Fortran Version 5.x math library.

`COURB.FON` Microsoft Fortran Version 5.x Courier graphics font file.

`CLGPOST.BAT` DOS batch file for compiling and linking the FORTRAN source codes to create the executable files.

`SUBDRV.FOR` Source code for the subroutines associated with passing data from the `FIRINPC` code to the plotting routines. These subroutines define the 25 plots, present the options to the user, and establish the upper and lower bounds on the horizontal (time) and vertical axes.

`SUBFIR.FOR` Source code for the subroutines associated with the actual fire compartment simulation. These routines perform file input/output, energy and mass balances, species and particulate concentrations, etc. This file contains the heart of the original mainframe version of the FIRIN computer code.

`SUBPLT.FOR` Source code for the subroutines associated with the logic of creating line drawings. The generic plotting subroutines are used to draw the curves, label the axes, draw tick marks, label curves, draw lines, etc.

`POST.FOR` Source code for the main program that operates the post processing package. This routine opens and reads the data files generated by a previous run of `FIRINPC`, stores the data in arrays, and passes it to the plotting routines. This file contains a copy of the `SUBROUTINE MSWIN`, which is the sole subroutine in the entire package that actually uses Microsoft FORTRAN Version 5.x graphics library calls.

`FIRINPC.FOR` Source code for the main FIRIN program that drives the fire compartment simulation package. This routine opens and reads the input data file generated by the FIRIN pre-processor, echoes certain of the information to an output file called `ECHO.DAT`, and begins the simulation. If interactive plotting is desired, the main program uses the subroutines in `SUBDRV.FOR` to pass information from FIRIN to the plotting routines. This file contains a copy of the `SUBROUTINE MSWIN`, which is the sole subroutine in the entire package that actually uses Microsoft FORTRAN Version 5.x graphics library calls.

`DECLARE.INC` This FORTRAN INCLUDE file contains the array and common block declarations related to the FIRIN simulation code. The majority of these variables are double precision.

`PLTPAR.INC` This FORTRAN INCLUDE file contains the array and common block declarations related to the graphics processing subroutines. The majority of these variables are single precision.

FIRACPC
Graphics Post-Processor Support
User's Guide

Dr. Edward Hensel, P.E.
Mechanical Engineering, Department 3450
New Mexico State University
Las Cruces, NM 88003-0001

July 1992

1 Introduction

FIRAC is a computer program used by DOE fire protection engineers to simulate the transient response of complete ventilation systems to fire induced transients. FIRAC has the ability to use the FIRIN code as the driving function or source term for the ventilation system response. FIRIN is a computer program used by DOE fire protection engineers to simulate hypothetical fire accidents in compartments at DOE facilities. This document describes how to install, test, and subsequently use the code POST, which is a program useful for graphically viewing the results of fire compartment (FIRIN) and full ventilation system (FIRAC) simulations. POST, is made available for reviewing the results of previous FIRIN or FIRAC simulations. POST uses the ASCII data files created by FIRIN and FIRAC.

2 PC Requirements and Installation

You must use an IBM 80386 PC or fully compatible alternative, with a VGA graphics board and monitor. You must have a math coprocessor, and use an 80386 or greater machine. If you wish to change the source code or recompile it as described in the programmer's reference, you will need to have a license for the SVS FORTRAN compiler. While the code can be run from a floppy disk, it is STRONGLY recommended that you use a hard disk.

This section of the user's guide contains instructions on how to install the POST code on your PC. First let us assume that your distribution diskette is to be placed in drive a:, and that your hard disk is designated as drive c:. To install the system, issue the following commands:

```
c:> cd c:\
c:> mkdir FIRACPC
c:> cd FIRACPC
```

```
c:\FIRACPC> copy a:*.* *.*
c:\FIRACPC> zoo -extract prob1.zoo *.*
```

The first command moves your current directory to the hard disk. Next, we create a new directory to receive the codes and data files, then move into that directory. We continue the installation by copying the files from the floppy disk to your hard disk. We use the disk pack utility, zoo, to unpack the results of FIRAC sample problem number 1 into an ascii format readable by the POST processing program. At this point, you should be ready to begin work and validation of the installation. You must verify that your system leaves its high memory area open, so that POST can use it. For example, if you have either of these commands in your CONFIG.SYS file, either remove them or comment them out (and then reboot you PC) prior to using POST. Be sure and make a backup copy of your CONFIG.SYS file prior to making these changes!

```
DOS=HIGH,UMB
DEVICE=C:\DOS\SMARTDRV.SYS 2500
```

In addition, you need to insure that your system will allow fortran to open enough data files for processing the results from FIRAC analyses. Make sure that you do have lines like this in your CONFIG.SYS file (with the appropriate hard disk specified for your machine):

```
STACKS=0,0
FCBS=20,0
BUFFERS=40
FILES=49
SHELL=C:\DOS|COMMAND.com /p /e:30000
```

Now that we have copied the files onto the hard disk, we need to verify the installation and learn how to use the programs. Let's start with the post-processing program, POST. Enter the command

```
c:\FIRACPC> POST
```

The program will begin execution by presenting the first level menu to you. From this menu, you may select screen graphics of the FIRIN or FIRAC responses. To begin, select option 1 from the menu:

```
Which option would you like?
1)  FIRIN Fire Compartment Graphics Review
2)  FIRAC Ventilating System Graphics Review
99) Exit to DOS .............................. 1
```

The program will next read the output data files from sample problem 1 provided with the distribution disk. After reading the data, POST will display a menu of plot options, as shown below.

```
Which kind of graphics would you like?

 1) Fire Compartment Thermal Effects
    11) Thermal Energy Balance            15) Hot Layer Thickness
    12) Mass Burn Rate                    16) Oxygen Volume Fraction
    13) Hot Layer Temperature             17) Fire Compartment Pressure
    14) Volumetric Flow Balance

 2) Smoke Source Term vs. Time         3) Radioactive Source Term vs. Time
    21) Smoke Accumulation                31) Radioactive Source Accumulation
    22) Hot Layer Loss Summary            32) Hot Layer Loss Summary
    23) Hot Layer Mass Balance            33) Hot Layer Mass Balance
    24) Cold Layer Loss Summary           34) Cold Layer Loss Summary
    25) Cold Layer Mass Balance           35) Cold Layer Mass Balance

 4) Particulate Size Distribution
    41) Radioactive Hot Layer             43) Smoke Hot Layer
    42) Radioactive Cold Layer            44) Smoke Cold Layer

99) RETURN to main menu.
```

You may now choose from among the 25 plotting options. Let us first view plot (1), which contains a summary of the fire compartment thermodynamics. To begin, type "1" and press the enter key on your keyboard. The program will now display a sequence of seven plots simultaneously on the screen. These seven plots correspond to each of the individual plot options labeled (11) through (17). You may review the mass flows, fire compartment pressure, temperatures, oxygen content, and energy balance for sample problem number 1 from the FIRIN manual. After you have finished looking at the plots, press the enter key to return to the main menu. If you next type "11" and press enter, you will see a full screen version of the thermal energy balance plot viewed previously in the fire compartment summary plot. The full screen version is more convenient for studying details of a particular response, while the summary plots are useful for understanding the relationships between events occurring in the fire compartment. After viewing plot (11), press the enter key again to return to the main option menu. To complete the installation verification of POST, you may review a combination of the other plotting options. After you finish viewing each plot, remember to press the enter key to return back to the main menu. To exit the post processing program, type "99" and press the enter key one more time.

Upon selecting FIRAC graphics, you will be presented with a second level menu as shown below:

You may now choose from among the FIRAC plotting options. Let us first view plot (5), which contains a summary of the nodal response of the system. You may plot the response of a maximum of 10 nodes at a time. After entering your choice for up to 10 nodes are of

```
Which kind of graphics would you like?

 5) FIRAC Node Summary                    6) FIRAC Branch Summary
    51) Node Pressures                       61) Branch Volume flows
    52) Node Temperatures                    62) Branch Mass flows
                                             63) Branch Damper Blade Angles

99) RETURN to main menu.
```

interest, POST will read the output data files froma previous run of the FIRAC code, and will gather the information for presentation. At the same time, you will be able to indicate a maximum of 10 branches of interest. You may review each of these plots as desrired, and when finished, return back to the main menu. Once at the main menu, you will again have an opportunity to perform FIRIN graphics or FIRAC graphics analysis.

If you wish to review the response of other than your originally selected nodes and branches, you need to go to the main menu, and re-enter the FIRAC submenu, so that you will be prompted to enter a new set of node and branch numbers.

Any time you wish to use the post processor, simply enter the command "POST" at the DOS prompt.

FAX (505) 665-2547 Bill Gregory

5 Progress Reports and Letters

Estimation and Control Laboratory
Mechanical Engineering, Dept. 3450
New Mexico State University
Box 30001/Las Cruces, New Mexico 88003-0001 USA
Telephone [505] 646-3501 Fax [505] 646-6111

March 9, 1992

Dr. Bill Gregory
Organization N6
Los Alamos National Laboratory
Los Alamos, NM

Dear Bill:

Per your request, I have prepared an outline of tasks that I feel should be completed during our ongoing effort related to FIRAC and creating a literature data base for DOE fire protection engineers. In this letter, I will begin with a review of the current situation on both aspects of the project, and then make some recommendations for future work. I will write the letter in a report format, so it will be somewhat easier to follow.

FIRAC: Current Status

As you are well aware, I have encountered a fairly pervasive error throughout FIRAC. The largest part of the problem is related to the particulate modeling aspects of the code. The error is the same in in all versions of FIRAC I have seen so far, including the most recent version FIRACN2.FOR.

The dimension of arrays referring to species concentrations are not consistently dimensioned. The problem is best illustrated by sample problem 1, with the number of `NGSPECE=13`. In routine `RGDATA` (called from `INPROC`) the code attempts to read

```
READ (IREAD,200) ISPEC(IS),IDSPEC(IS,1),IDSPEC(IS,2)
```

with the value of `IS = 10`. However, the arrays are only dimensioned to a size of

```
COMMON /GASVLT/ ISPEC(5), IDSPEC(5,2)
```

In addition, `RGDATA` calls `CLEAR2` to zero out columns of arrays in

```
COMMON /GASPT/ GFLUX(100,5),GFLUXI(100,5),GCON(100,5)
```

which do not even exist! This results in memory mapping errors which call into serious question the validity of the results obtained with the code. Indeed, when I compile the code with full debugging options implemented on the VMS compiler, it becomes clear that array access violations are common throughout the code. There are also some errors with type mismatches between the calling routine and the called routines on several argument lists. For example, routine clear is used to initialize integer and real arrays to zero. This seems to work fine on the CRAY FORTRAN compiler, but causes errors on FORTRAN 77 compilers.

I feel that these errors are critical issues that must be resolved before the output, particularly as related to the particulate model, from the FIRAC code can be trusted. The errors are memory mapping errors of a nature that may very well not be detected by many FORTRAN compilers. I believe that it is essential to perform an in-depth verification of the code as soon as possible. It appears that many changes have been implemented in the code over a period of time, and the code may not have been fully verified (analytically) following completion of many such changes. Of particular concern are subtle errors associated with changing the dimensions of arrays and problem sizes.

I have now worked with the FIRAC code, in one form or another, for several years. I have been heavily involved with the inner workings of the code for about 4 years now. I was actively involved in much of the experimental support of the code here at NMSU, and was a code user long before getting involved with its inner workings. Altogether, I have nearly ten years of experience with the FIRAC, TORAC, and EXPAC packages.

FIRAC: Proposed Directions

Based on my past experience with FIRAC and several other large simulation codes, I feel that it is necessary to address several issues related to the code. The items that are essential to insuring a robust and reliable code are:

1. FIRAC (as well as FIRIN, TORAC, and EXPAC) have all grown over the years. They were originally developed on large CRAY-class supercomputers and mainframes with severe memory limitations. As computer technology has advanced, the capabilities of FIRAC have been increased to allow larger models, more species, various materials, etc. However, it is clear all of the changes have not been accompanied by a full suite of verification tests at each step. I have located a number of errors in the current version of the code as related to memory allocation and array mapping. I recommend that, as an *absolute minimum*, it is necessary to review the entire code for such errors, and to revise the code to guard against the future introduction of such errors. As a first step, this should consist of removing all hard-wired array dimensions from declaration statements, and replace all dimensions with fortran parameters, which will dramatically enhance code reliability. I am confident that this single step will help us identify several errors in the code. Please note that these changes are an effort to modernize the code, using software and hardware technology that simply was not available at the time FIRAC was originally written. Using my own past productivity as a guide, I estimate that this effort, carefully done to insure reliability, would take about 6 man-months to complete.
2. As an interim solution, the common blocks should be moved to INCLUDE files. Combined with the use of PARAMETERS, this will improve code reliability. I estimate that this effort could be completed in conjunction with item 1.
3. Each subroutine *must* have some level of verification (by analytical means). With the combined systems of engineering units used between the FIRAC and FIRIN codes, I

am afraid that there may be some numerical inconsistancies.

4. After all of the changes, the code must be re-validated (against existing experimental data where available). Phil Smith would be a good person to speak with about the experimental validation aspects of the project.
   I recommend that the software modifications should be done by one individual, or by a very small team of engineers working closely together. It is my opinion that many of the past problems with FIRAC have arisen out of the fact that there have been "too many cooks" and that no single person now has a clear picture of the entire code. Thus, the engineers must have a good understanding of the fire modeling aspects of the code as well as the software engineering and programming issues.
   Rather than having CRAY, a SUN, and PC versions of the code, we will have a *single* set of FORTRAN source modules that can be recompiled and linked together on various machines. This would eliminate our current problem of one version apparently running successfully, while another version can not. More powerful computers will have access to all of the code modules, while personal computers will have access only to those modules that can be realistically executed on a given machine. For example, a fire compartment module based on FIRIN may be appropriate to run on a PC or workstation. A fire compartment model with full sprinkler support may require a 3-d transient simulation, and could only be run on a large supercomputer.

5. I recommend that all common blocks be removed from the code, in favour of argument lists. This will allow us to identify all appropriate code dependencies, and to establish a formal program verification methodology. We can then fully debug one module at a time, without worrying that we introduce an error in an auxiliary code segment. The current code structure makes it virtually impossible to perform formal program validation.

6. I recommend that the code make extensive use of existing scientific and engineering software libraries available through U.S. government sources. For example, the DOE SLATEC library contains several routines that have undergone extensive validation within the DOE complex and are already maintained by a professional staff. I feel that using the existing libraries will allow us an opportunity to provide users with alternative solution strategies that may be more appropriate for a given problem.

7. I recommend that the code be re-written in FORTRAN 90. This new implementation of the FORTRAN standard is a superset of the existing FORTRAN 77 standard. The new language will allow us to take advantage of improved memory management routines, dynamic array allocation to improve run-time requirements, and will enable us to exploit the power of the new generation of parallel supercomputers. I have a great deal of experience with FORTRAN 90, and I am quite impressed by the additional power it brings to engineering application software. If we first modularize the existing code, then the conversion can be done on a module-by-module basis. That way, we always have an existing code and on-going capability – we don't want to go through a period of time

where the fire protection engineers have no tools to work with. We can perform the conversion by starting with the most cpu and memory intensive modules, and continue on until the conversion is complete. All the way, we will validate each module on a wide variety of machines (PCs; SUN, IBM, DEC, and SGI workstations; Mainframes; and CRAY, CM5, and INTEL parallel supercomputers). I believe we are in a unique position to get FIRAC and related modules running on the newest parallel supercomputers in the DOE complex. Indeed, our group is already using the DOE machines at Los Alamos and Sandia, as well as an Air Force machine at NPAC and an Army machine at Minnesota. We have developed several finite difference and finite element codes on these machines, and are beginning to look at a particulate transport code as well.

8. A suite of analytical verification and experimental validation problems must be established. These tools must be used to verify and validate each individual module and the complete code.
9. The user interface for the code should be improved. We have taken a big step forward in the last six months, with the pre- and post-processor capabilities. We must continue this effort, and must add improved graphics capabilities as we add more powerful modeling modules. We might wish to consider a more standard approach to the user interface. The pre-processor appears to be heavily tied to a PC environment, while I already have the post-processor running on a variety of machines. As an example of future user-interface needs, a 3-d transient fire compartment module should be supported by fringe plot graphics capabilities, 3-d rotations, semi-transparent images, and particulate transport visualization. I am currently involved with a group looking at particulate transport for the Kuwait smoke plume – I think many of the visualization concepts we employ there would apply well to the FIRAC code.

DATABASE: Current Status

We have completed our analysis of the NTIS database information that you provided to us on floppy disks, and we are now studying the FIREDOC database. It is quite apparent that we will generate a large quantity of information. A major issue we must consider is the best medium by which to deliver this vast amount of information to DOE fire protection engineers. I doubt whether a hard copy listing of bibliographic citations or abstracts will be of much use to DOE engineers. With the NTIS data alone, the bibliographic citations already fill two 3-ring binders. This volume is certain to increase dramatically as we study additional data sources. Our preliminary report indicates the magnitude of the data base that we are looking at.

DATABASE: Proposed Directions

I propose that we establish a context sensitive data base that engineers can interact with to get data that specifically matches their needs. This information should be accessed easily, either through their local PC, Mac, or workstation or via telephone or the internet to a remote central data vault. A major advantage to this approach is that the central repository can be responsible for maintaining a current list of relevant information. The NIST FIREDOC database already accomplishes much of this mission. It would be worth considering providing funds to NIST to expand their capability and have it meet the needs of DOE. If we determine it best to expand NIST capabilities, I suggest they be supported to improve the telecommunications and internet access, provide better remote host support, and improve their data download capabilities to include something along the lines of KERMIT data compression or email delivery of abstracts. The National Science Foundation runs the STIS database with such capabilities for science and technology programs through NSF. Several commercial companies such as COMPENDEX have general engineering literature databases. It would be wise to model the DOE approach after such an industry standard.

If it is deemed inappropriate to expand the capabilities of the NIST FIREDOC database, then the next logical alternative would be to establish a DOE clearing house for fire protection documentation, abstracts, and technical data. As a faculty member at NMSU, I am sure we could include such a capability under the auspices of the DOE WERC program already in place. We could provide on-line support to remote users via telephone lines and the internet. I believe our congressional delegation would be supportive of a capability that makes use of the New Mexico TECHNET. We could provide off-line support to DOE users in the form of hard-copy (where we do the search for them and mail them the results), soft-copy electronic delivery via email, magnetic media, or CD ROM. We could support a graduate student to maintain an annual update of the data base, and provide professional staff support for a hot-line support center. We would provide assistance to DOE users at DOE sites to provide them with turnkey systems for their own needs. It would be natural to include assistance with fire data, abstracts, and perhaps even to study the feasibility of providing technical support on the fire codes such as FIRAC.

Summary

Bill, I hope this letter has served to put several topics on the table for discussion. I feel strongly that several technical issues must be resolved with the FIRAC code. I believe further that we can take this as an opportunity to modernize the code and make it even more useful to the DOE community and beyond. I would like to stay heavily involved with the code development and modernization – I believe NMSU brings several unique capabilities to the project. We are always interested in working with the national labs as part of the DOE team. I hope you appreciate the fact that we are milestone conscious, and are very aware of your time constraints.

We should study several delivery mechanisms for the fire literature database. It is important that we pick a delivery mechanism that will be accepted by the user community, and also an approach that will lend itself to economical maintenance and update. We would like to remain involved with this aspect of the project, as I believe it to be an excellent opportunity for MS level engineers to become familiar with an important field of technology.

Please feel free to let me know what you think of these ideas. I would be happy to sit down and discuss them with you and your colleagues in more detail. I am looking forward to hearing from you.

Sincerely,

Edward Hensel
Associate Professor & Director of ECL
E-Mail: EHensel@helen.nmsu.edu

Estimation and Control Laboratory
Mechanical Engineering, Dept. 3450
New Mexico State University
Box 30001/Las Cruces, New Mexico 88003-0001 USA
Telephone [505] 646-3501 Fax [505] 646-6111

March 20, 1992

Dr. Bill Gregory
Organization N6
Los Alamos National Laboratory
Los Alamos, NM

Dear Bill:

Per our recent telephone conversation, I have prepared a list of items I feel we should address during the coming two months as we prepare for an evaluation release of the FIRAC code and related pre/post processors. The list here represents the items I feel can be accomplished within our 2 month time frame, and will give the best possible product during that period.

The task I suggest completing within two months are:

1. I recommend that, as an *absolute minimum*, it is necessary to review the entire code for memory mapping and array dimension errors, and to revise the code to guard against the future introduction of such errors. For a two month project, I recommend placing all variables in common blocks, with a single include file, and changing all array declarations to rely upon parameter. This is the general idea I used in the FIRIN module and the post-processor module. I believe this effort should be led by the team from Hanford.
2. A formal argument list should be passed into and out of the FIRIN code. We should try to use the newest version of FIRINPC, with separate source code modules. This may not be achievable within two months. I believe this effort should be led by the team from Hanford, with me providing support on my implementation of and changes to the FIRIN code.
3. The particulate graphics support modules should be completed in the post processor. While I have made substantive progress here, I will not feel comfortable with it until I can test the code with several sets of output data. Right now I can only use "dummy data." I will address several smaller issues such as axis scaling, labeling bugs, and things of that nature. There are no major hurdles with the post-processing code at this point, but a great deal of time and care must go into polish, checking for bugs, and trying to make the code more robust to user inputs. I will be responsible for this effort. I will not be able to complete this effort until Item 1 is completed. I estimate I will need a MINIMUM of three weeks with the working FIRAC code to feel comfortable about releasing the graphics post processor.
4. The FIREDOC database work is nearing completion. I expect that the notebooks will be completed within a week after they are returned to me. My students, Wayne and Rod, are responsible for this effort.

5. We should compile several test problems using the pre-processor. These might be problems from the FIRAC manual. My students can contribute to this effort. They will likely need guidance from you.
6. I would recommend a short 1-2 day training session or video tape for the evaluators of the code. The training session could go through how to install the software on their system and run a sample problem. Also, the manuals need to be either combined together or at least made avaliable to the evaluators. It seems like you might be the best person to lead this effort.

Sincerely,

Edward Hensel
Associate Professor & Director of ECL
E-Mail: EHensel@helen.nmsu.edu

Mechanical Engineering Department – New Mexico State University
P. O. Box 30001, Dept. 3450, Las Cruces, NM 88003
TEL: (505) 646-3501 FAX: (505) 646-6111

FAX COVER SHEET

DATE: 20 March 1992

TO: Bill Gregory FAX NO: 665-2897

FROM: Ed Hensel
Voice 646-3501
FAX 646-6111

PAGES FOLLOWING: 2

RE: 2 month effort details.

MESSAGE: I'm on my way to a PhD exam. I'll call when I get out or speak with you on Monday.
Ed

Mechanical Engineering Department – New Mexico State University
P. O. Box 30001, Dept. 3450, Las Cruces, NM 88003
TEL: (505) 646-3501 FAX: (505) 646-6111

FAX COVER SHEET

DATE: 1 April 1992

TO: Stan Clybrook FAX NO: 509-943-6617

FROM: Ed Hensel

PAGES FOLLOWING: 10

RE: File read subroutines for my post-processor program. I would like to speak with you about modifying FIRAC to produce an ascii format

MESSAGE: File FORT.13 that would be readable by this subroutine "Get_firac."

Ed Hensel
(505) 646-2011

```
-----------------------------------------------------------------

SUBROUTINE GET_FIRIN ( NTIME )

-----------------------------------------------------------------

THIS ROUTINE RETRIEVES INFORMATION GENERATED BY FIRIN AND STORED
IN VARIOUS ASCII DATA FILES

-----------------------------------------------------------------

INCLUDE 'PLTPAR.INC'
REAL*4 R4TFIN, R4DELT, ROW(11)
REAL*4 R4COMP(11), R4SMOK(23), R4RADS(23)
INTEGER*4 I4NPLT, I4INIT, I4FINI, I4GRAF
INTEGER*4 I4TIME
CHARACTER*1 CANS
CHARACTER*132 CLINE

-----------------------------------------------------------------

Initialization Block - Recall the FIRIN generated data from disk.
Open data files that were created as output from FIRIN
Read the available data from each file, watching for header info
  and informational messages printed to the files.
Pad any missing data for times when cold layer does not exist.
Close each data file as soon as possible.

WRITE(*,*) ' Starting to read FIRIN output data files ... '

Read in the fire compartment thermal effects from PRINT.DAT

OPEN ( UNIT=10, FILE='PRINT.DAT', STATUS='OLD' )
NTIME = 0
DO 9 I = 1, 8
  READ(10,100) CLINE
9     CONTINUE
0     READ(10,100, ERR=10,END=11) CLINE
READ(CLINE,270,ERR=10,END=10) TSTEP, PCOMP, FMO2,ZHL,VIF,VOF,
.             THLUC, TMASSR, TQACTR, QLOSSR, TQNETR
  THIS LOGIC IS REQUIRED SINCE MICROSOFT FORTRAN DOES NOT SUPPORT
  THE USE OF LIST DIRECTED READS ON INTERNAL UNITS. THIS WILL
  AVOID ASSIGNING ZERO VALUES TO A COMPLETELY BLANK LINE IN THE
  INPUT FILES.
  IF ( NTIME .GT. 1 .AND. TSTEP .EQ. 0.0 ) GO TO 10
  NTIME = NTIME + 1
  TIMVEC(NTIME)    =                TSTEP     ! Seconds
  YVECCP(NTIME,1) = 14.69595 * PCOMP     ! (atm) to (psi)
  YVECO2(NTIME,1) =                FMO2      ! volume fraction
  YVECHL(NTIME,1) = 3.28084   * ZHL       ! (m) to (ft)
  YVECQQ(NTIME,1) = 2118.88   * VIF       ! (m^3/sec) to (cfm)
  YVECQQ(NTIME,2) = 2118.88   * VOF
  YVECHT(NTIME,1) =                THLUC     ! degrees (F)
  YVECBR(NTIME,1) = 7.93667   * TMASSR    ! (g/s) to (lbm/hr)
  YVECKW(NTIME,1) = 3412.13   * TQACTR    ! (kw)  to (btu/hr)
  YVECKW(NTIME,2) = 3412.13   * QLOSSR
  YVECKW(NTIME,3) = 3412.13   * TQNETR
  GO TO 10
1     WRITE(*,*) NTIME,' time steps of data were read from PRINT.DAT'
```

```
      CLOSE(UNIT=10)

      SMOKE SOURCE TERMS COLD AND HOT LAYERS...

      CVTMAS = 32.17 / 14593.9   ! convert (g) to (lbm)

      Read in the cold layer smoke terms from SMKCL.DAT

      NT = 0
      OPEN ( UNIT= 7, FILE='SMKCL.DAT', STATUS='OLD',ERR=14 )
      READ(7,100) CLINE
      READ(7,100) CLINE
      READ(7,100) CLINE
      READ(7,100) CLINE
      READ(7,100) CLINE
}     READ(7,*,ERR=13,END=14)TDUMMY,BSTART,HLSETL,BLOSS,BFINI,
     .          FRAC1, FRAC2, FRAC3
        IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 13
        NT = NT + 1
        YVSCLG(NT,1) = CVTMAS * BSTART ! Airborne at t0
        YVSCLG(NT,2) = CVTMAS * HLSETL ! HL Settling
        YVSCLG(NT,3) = CVTMAS * BLOSS  ! Total Losses
        YVSCLG(NT,4) = CVTMAS * BFINI  ! Airborne at t
        YVSCLF(NT,1) =          FRAC1  ! Fraction settling
        YVSCLF(NT,2) =          FRAC2  ! Fraction out vent
        YVSCLF(NT,3) =          FRAC3  ! Fraction to plume
        GO TO 13
4       CONTINUE
        IF ( NT .LT. NTIME ) THEN
          WRITE(*,*) NTIME-NT,' missing data-pairs in SMKCL.DAT *** '
          PAD WITH ZEROS FOR TIMES WHEN THERE IS NO LONGER A COLD LAYER.
          DO N = NT+1, NTIME
            YVSCLG(N,1) = 0.0 ! Airborne at t0
            YVSCLG(N,2) = 0.0 ! HL Settling
            YVSCLG(N,3) = 0.0 ! Total Losses
            YVSCLG(N,4) = 0.0 ! Airborne at t
            YVSCLF(N,1) = 0.0 ! Fraction settling
            YVSCLF(N,2) = 0.0 ! Fraction out vent
            YVSCLF(N,3) = 0.0 ! Fraction to plume
          END DO
        END IF
      CLOSE(UNIT=7)

      Read in the hot layer smoke terms from SMKHL.DAT

      NT = 0
      OPEN ( UNIT=15, FILE='SMKHL.DAT', STATUS='OLD',ERR=16 )
      READ(15,100) CLINE
      READ(15,100) CLINE
      READ(15,100) CLINE
      READ(15,100) CLINE
      READ(15,100) CLINE
.5    READ(15,*,ERR=15,END=16) TDUMMY,ASTART,AGEN,APLUME,ALOSS,
     .          AFINI, FRAC1, FRAC2, FRAC3, FRAC3, FRAC4
        IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 15
        NT = NT + 1
!       HOT LAYER SMOKE TERMS
        YVSHLG(NT,1) = CVTMAS * ASTART ! Airborne at t0
        YVSHLG(NT,2) = CVTMAS * AGEN   ! Generated by fire
        YVSHLG(NT,3) = CVTMAS * APLUME ! Entrained w/Plume
```

```
      YVSHLG(NT,4) = CVTMAS * ALOSS  ! Total Losses
      YVSHLG(NT,5) = CVTMAS * AFINI  ! Airborne at t
      YVSHLF(NT,1) = FRAC1   ! Fraction settling
      YVSHLF(NT,2) = FRAC2   ! Fraction B.Duffus
      YVSHLF(NT,3) = FRAC3   ! Fraction Thermoph
      YVSHLF(NT,4) = FRAC3   ! Fraction Diffusio
      YVSHLF(NT,5) = FRAC4   ! Fraction Out Vent
      GO TO 15
      CONTINUE
      IF ( NT .LT. NTIME ) THEN
        WRITE(*,*) NTIME-NT,' missing data-pairs in SMKHL.DAT *** '
        PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
        DO N = NT+1, NTIME
          DO J = 1, 5
            YVSHLG(N,J)  = 0.0
            YVSHLF(N,J) = 0.0
          END DO
        END DO
      END IF
    CLOSE(UNIT=15)

    Read in the smoke accumulation terms from ASMK.DAT

    NT = 0
    OPEN ( UNIT=14, FILE='ASMK.DAT' , STATUS='OLD',ERR=19 )
    READ(14,100) CLINE
    READ(14,100) CLINE
    READ(14,100) CLINE
    READ(14,100) CLINE
8     READ(14,*,ERR=18,END=19) TDUMMY,FLOOR,WALL,ROOF,VENTS,ENVIR
      IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 18
      NT = NT + 1
      SMOKE ACCUMULATION TERMS
      YVSACC(NT,1) = CVTMAS * FLOOR
      YVSACC(NT,2) = CVTMAS * WALL
      YVSACC(NT,3) = CVTMAS * ROOF
      YVSACC(NT,4) = CVTMAS * VENTS
      YVSACC(NT,5) = CVTMAS * ENVIR
      GO TO 18
9     CONTINUE
      IF ( NT .LT. NTIME ) THEN
        WRITE(*,*) NTIME-NT,' missing data-pairs in ASMK.DAT *** '
        PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
        DO N = NT+1, NTIME
          DO J = 1, 5
            YVSACC(N,J) = 0.0
          END DO
        END DO
      END IF
    CLOSE(UNIT=14)

    RADIOACTIVE SOURCE TERM

    Read in the cold layer radioactive terms from RSTCL.DAT

    NT = 0
    OPEN ( UNIT= 4, FILE='RSTCL.DAT', STATUS='OLD',ERR=21 )
    READ(4,100) CLINE
    READ(4,100) CLINE
    READ(4,100) CLINE
```

```
     READ(4,100) CLINE
     READ(4,100) CLINE
0      READ(4,*,ERR=20,END=21) TDUMMY,BSTART,HLSETL,BLOSS,BFINI,
   .         FRAC1, FRAC2, FRAC3
       IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 20
       NT = NT + 1
       YVRCLG(NT,1) = CVTMAS * BSTART ! Airborne at t0
       YVRCLG(NT,2) = CVTMAS * HLSETL ! HL Settling
       YVRCLG(NT,3) = CVTMAS * BLOSS  ! Total Losses
       YVRCLG(NT,4) = CVTMAS * BFINI  ! Airborne at t
       YVRCLF(NT,1) =          FRAC1  ! Fraction settling
       YVRCLF(NT,2) =          FRAC2  ! Fraction out vent
       YVRCLF(NT,3) =          FRAC3  ! Fraction to plume
       GO TO 20
1      CONTINUE
       IF ( NT .LT. NTIME ) THEN
         WRITE(*,*) NTIME-NT,' missing data-pairs in RSTCL.DAT *** '
         PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
         DO N = NT+1, NTIME
           YVRCLG(N,1) = 0.
           YVRCLG(N,2) = 0.
           YVRCLG(N,3) = 0.
           YVRCLG(N,4) = 0.
           YVRCLF(N,1) = 0.
           YVRCLF(N,2) = 0.
           YVRCLF(N,3) = 0.
         END DO
       END IF
     CLOSE(UNIT=4)

     Read in the hot layer radioactive terms from RSTHL.DAT

     NT = 0
     OPEN ( UNIT= 3, FILE='RSTHL.DAT', STATUS='OLD', ERR=26 )
     READ(3,100) CLINE
     READ(3,100) CLINE
     READ(3,100) CLINE
     READ(3,100) CLINE
     READ(3,100) CLINE
5      READ(3,*,ERR=25,END=26) TDUMMY,ASTART,AGEN,APLUME,ALOSS,
   .         AFINI, FRAC1, FRAC2, FRAC3, FRAC3, FRAC4
       IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 25
       NT = NT + 1
       HOT LAYER SMOKE TERMS
       YVRHLG(NT,1) = CVTMAS * ASTART ! Airborne at t0
       YVRHLG(NT,2) = CVTMAS * AGEN   ! Generated by fire
       YVRHLG(NT,3) = CVTMAS * APLUME ! Entrained w/Plume
       YVRHLG(NT,4) = CVTMAS * ALOSS  ! Total Losses
       YVRHLG(NT,5) = CVTMAS * AFINI  ! Airborne at t
       YVRHLF(NT,1) = FRAC1   ! Fraction settling
       YVRHLF(NT,2) = FRAC2   ! Fraction B.Duffus
       YVRHLF(NT,3) = FRAC3   ! Fraction Thermoph
       YVRHLF(NT,4) = FRAC3   ! Fraction Diffusio
       YVRHLF(NT,5) = FRAC4   ! Fraction Out Vent
       GO TO 25
5      CONTINUE
       IF ( NT .LT. NTIME ) THEN
         WRITE(*,*) NTIME-NT,' missing data-pairs in RSTHL.DAT *** '
         PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
         DO N = NT+1, NTIME
```

```
        DO J = 1, 5
          YVRHLG(N,J)  = 0.0
          YVRHLF(N,J) = 0.0
        END DO
      END DO
    END IF
  CLOSE(UNIT=3)

  Read in the RADIOACTIVE accumulation terms from ARAD.DAT

  NT = 0
  OPEN ( UNIT=12, FILE='ARAD.DAT' , STATUS='OLD',ERR=29 )
  READ(12,100) CLINE
  READ(12,100) CLINE
  READ(12,100) CLINE
  READ(12,100) CLINE
8   READ(12,*,ERR=28,END=29) TDUMMY,FLOOR,WALL,ROOF,VENTS,ENVIR
    IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 28
    NT = NT + 1
    RADIOACTIVE ACCUMULATION TERMS
    YVRACC(NT,1) = CVTMAS * FLOOR
    YVRACC(NT,2) = CVTMAS * WALL
    YVRACC(NT,3) = CVTMAS * ROOF
    YVRACC(NT,4) = CVTMAS * VENTS
    YVRACC(NT,5) = CVTMAS * ENVIR
    GO TO 28
9   CONTINUE
    IF ( NT .LT. NTIME ) THEN
      WRITE(*,*) NTIME-NT,' missing data-pairs in ARAD.DAT *** '
      PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
      DO N = NT+1, NTIME
        DO J = 1, 5
          YVRACC(N,J)  = 0.0
        END DO
      END DO
    END IF
  CLOSE(UNIT=12)

  RADIOACTIVE SOURCE PARTICLE DISTRIBUTION IN THE HOT LAYER

  NT = 0
  OPEN ( UNIT= 8, FILE='RSZHL.DAT', STATUS='OLD',ERR=31 )
  READ(8,100,END=31) CLINE
  READ(8,100,END=31) CLINE
  READ(8,100,END=31) CLINE
  READ(8,100,END=31) CLINE
  READ(8,*) ( DPART4(I), I = 1, 11 )
0   READ(8,*,ERR=30,END=31) TDUMMY, (ROW(I), I=1,11)
    IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 30
    NT = NT + 1
    DO I = 1, 11
      YVRHLZ(NT,I) = CVTMAS * ROW(I)
    END DO
    GO TO 30
1   CONTINUE
    IF ( NT .LT. NTIME ) THEN
      WRITE(*,*) NTIME-NT,' missing data-pairs in RSZHL.DAT *** '
      PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
      DO N = NT+1, NTIME
        DO I = 1, 11
```

```
            YVRHLZ(N,I) = 0.0
          END DO
        END DO
      END IF
    CLOSE(UNIT=8)

    RADIOACTIVE SOURCE PARTICLE DISTRIBUTION IN THE COLD LAYER

    NT = 0
    OPEN ( UNIT= 9, FILE='RSZCL.DAT', STATUS='OLD',ERR=41 )
    READ(9,100,END=41) CLINE
    READ(9,100,END=41) CLINE
    READ(9,100,END=41) CLINE
    READ(9,100,END=41) CLINE
    READ(9,100,END=41) CLINE
0     READ(9,*,ERR=40,END=41) TDUMMY, (ROW(I), I=1,11)
      IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 40
      NT = NT + 1
      DO I = 1, 11
        YVRCLZ(NT,I) = CVTMAS * ROW(I)
      END DO
      GO TO 40
1     CONTINUE
      IF ( NT .LT. NTIME ) THEN
        WRITE(*,*) NTIME-NT,' missing data-pairs in RSZCL.DAT *** '
        PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
        DO N = NT+1, NTIME
          DO I = 1, 11
            YVRCLZ(N,I) = 0.0
          END DO
        END DO
      END IF
    CLOSE(UNIT=9)

    SMOKE SOURCE PARTICLE DISTRIBUTION IN THE HOT LAYER

    NT = 0
    OPEN ( UNIT=11, FILE='SSZHL.DAT', STATUS='OLD',ERR=51 )
    READ(11,100,END=51) CLINE
    READ(11,100,END=51) CLINE
    READ(11,100,END=51) CLINE
    READ(11,100,END=51) CLINE
    READ(11,100,END=51) CLINE
0     READ(11,*,ERR=50,END=51) TDUMMY, (ROW(I), I=1,11)
      IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 50
      NT = NT + 1
      DO I = 1, 11
        YVSHLZ(NT,I) = CVTMAS * ROW(I)
      END DO
      GO TO 50
1     CONTINUE
      IF ( NT .LT. NTIME ) THEN
        WRITE(*,*) NTIME-NT,' missing data-pairs in SSZHL.DAT *** '
        PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
        DO N = NT+1, NTIME
          DO I = 1, 11
            YVSHLZ(N,I) = 0.0
          END DO
        END DO
      END IF
```

```
      CLOSE(UNIT=11)

      SMOKE SOURCE PARTICLE DISTRIBUTION IN THE COLD LAYER

      NT = 0
      OPEN ( UNIT=13, FILE='SSZCL.DAT', STATUS='OLD',ERR=61 )
      READ(13,100,END=61) CLINE
      READ(13,100,END=61) CLINE
      READ(13,100,END=61) CLINE
      READ(13,100,END=61) CLINE
      READ(13,100,END=61) CLINE
0       READ(13,*,ERR=60,END=61) TDUMMY, (ROW(I), I=1,11)
        IF ( NT .GT. 1 .AND. TDUMMY .EQ. 0.0 ) GO TO 60
        NT = NT + 1
        DO I = 1, 11
          YVSCLZ(NT,I) = CVTMAS * ROW(I)
        END DO
        GO TO 60
1       CONTINUE
        IF ( NT .LT. NTIME ) THEN
          WRITE(*,*) NTIME-NT,' missing data-pairs in SSZCL.DAT *** '
          PAD WITH ZEROS FOR TIMES WHEN THERE IS NO DATA
          DO N = NT+1, NTIME
            DO I = 1, 11
              YVSCLZ(N,I) = 0.0
            END DO
          END DO
        END IF
      CLOSE(UNIT=13)

      ******************************************************************

      WRITE(*,*) '   FIRIN starting time ..... ', TIMVEC(1)
      WRITE(*,*) '   FIRIN finish time ....... ', TIMVEC(NTIME)

      RETURN
00    FORMAT(A)
45    FORMAT(1H ,F8.2,12(1X,E9.3) )
70    FORMAT(1X,F9.2,10(1X,F11.4))
10    FORMAT(1H ,F9.2,5(3X,E9.2),5(1X,F9.4))
20    FORMAT(1H ,F9.2,4(3X,E9.2),3(1X,F9.4))
      END

      ------------------------------------------------------------------

      SUBROUTINE GET_FIRAC ( NTIME )

      ------------------------------------------------------------------

      REWIND THE SCRATCH FILE, AND INSERT SELECTED DATA INTO SINGLE
      PRECISION PLOTTING ARRAYS. IN THE PROCESS, CONVERT THE DATA TO
      EITHER SI OR ENGLISH UNITS AS REQUESTED BY THE OPERATOR.

      ------------------------------------------------------------------

      INPUTS:
      I*4 KFSI      FLAG FOR OUTPUT DISPLAY UNITS
                    0 PRESENT RESULTS IN ENGLISH UNITS
                    1 PRESENT RESULTS IN SI UNITS
      I*4 NTOUT     NUMBER OF TIMES PRINTED TO ISCR BY FIRAC
```

```
    I*4 ISCR      LUN OF SCRATCH BINDARY FILE GENERATED BY FIRAC
    I*4 NSPECES   NUMBER OF PARTICULATE SPECIES IN FIRAC ANALYSIS
    I*4 NNODES    NUMBER OF NODES IN FIRAC ANALYSIS
    I*4 NBRCH     NUMBER OF BRANCHES IN FIRAC ANALYSIS
COM I*4 NNPLT     # OF NODES REQUESTED FOR THE PLOTS
COM I*4 NNPLTP    POINTER ARRAY FOR NODES REQUESTED
COM I*4 NBPLT     # OF BRANCHES REQUESTED FOR THE PLOT
COM I*4 NBPLTP    POINTER ARRAY FOR BRANCHES REQUESTED
    PARAMETERS:
COM I*4 MTIME     MAX NUMBER OF PLOTTING TIMES
COM I*4 MCURVE    MAX NUMBER OF BRANCHES OR NODES TO BE DISPLAYED
COM I*4 MSPECES   MAX NUMBER OF PARTICULATE SPECIES IN THE ANALYSIS
    OUTPUTS:
        RUMCRV
        BRHCRV
    LOCAL VARIABLES
    R*4  QTOSI
    R*4  PTOSI
    R*4  CVTEMP   FROM (F) TO (C)
    R*4  PLBTSI
    R*4  TZERK    ZERO (C) FROM KELVIN
    R*4  PZERO    (psia)
    R*4  XQSI
    R*4  XPSI
    R*4  PAMBS
    I*4  NT       TIME STEP INCREMENT
    I*4  I,J,K    DO LOOP INCREMENTS

    ------------------------------------------------------------------

    INCLUDE 'PLTPAR.INC'
    REAL P(100), Q(100), MFLOW(100), T(100), THETA(100)
    REAL DFLUX(5), PMOF(5), WMASS(5), SRCE(5), SINK(5), Y(5)

    CONVERSION FACTORS

    DATA QTOSI  / 4.71947e-4 /
    DATA PTOSI  /  248.84    /
    DATA CVTEMP /    1.8     /    ! (F/C)
    DATA PLBTSI / 6894.757   /
    DATA TZERK  /  273.15    /    ! ZERO (C) FROM KELVIN
    DATA PZERO  /   14.7     /    ! (psia)
    XQSI  = 1.0    / QTOSI
    XPSI  = 1.0    / PLBTSI
    PAMBS = PZERO * PLBTSI
    REWIND ISCR

    DECIDE ON THE PLOTTING INCREMENT.

    ISCR     = 15
    OPEN(UNIT=ISCR,FILE='FIRAC.DAT',STATUS='UNKNOWN')
    .      FORM='UNFORMATTED')

    NNODES   = 18
    NBRCH    = 17
    NSPECES = 0

    DECIDE WHICH INFORMATION TO STORE.

00  FORMAT(1X,'How many nodes do you wish to review? ....... ',$ )
```

```
10    FORMAT(1X,'Enter desired node numbers: ................. ',$ )
40    FORMAT(1X,'How many branches do you wish to review? .... ',$ )
50    FORMAT(1X,'Enter desired branch numbers: .............. ',$ )

      REQUEST CHANGES IN NODE NUMBERS

      WRITE(*,100)
      READ (*,*)   NNPLT
      WRITE(*,110)
      READ (*,*)  (NNPLTP(NN),NN=1,NNPLT)

      REQUEST CHANGES IN BRANCH NUMBERS

      WRITE(*,140)
      READ (*,*)   NBPLT
      WRITE(*,150)
      READ (*,*)  (NBPLTP(NB),NB=1,NBPLT)

      LOOP OVER ALL TIMES TO GET THE DATA FROM THE SCRATCH FILE

      KOUNT = 0
      IER = 0
      WRITE(*,*) ' Starting to read FIRAC output data files ... '
      DO WHILE ( IER .EQ. 0 )

        Retrieve node and branch information for flow/energy parameters

        READ (ISCR,*,END=1440,IOSTAT=IER)
     .                          TIME, ( P(I),      I = 1, NNODES ),
     .                                ( Q(J),      J = 1, NBRCH  ),
     .                                ( MFLOW(J), J = 1, NBRCH  ),
     .                                ( T(I),      I = 1, NNODES ),
     .                                ( THETA(I), I = 1, NBRCH  )
         IF (NSPECES.GT.0) THEN
C
C           Retrieve the node and branch information for each species
C
            DO K=1,NSPECES
C             DISCARD THE INFO AS SOON AS IT IS READ...
              READ (ISCR,*,END=1440,IOSTAT=IER)
     .                                  ( DFLUX(I), I = 1, NBRCH  ),
     .                                  ( PMOF(I),  I = 1, NBRCH  ),
     .                                  ( WMASS(I), I = 1, NBRCH  ),
     .                                  ( SRCE(I),  I = 1, NNODES ),
     .                                  ( SINK(I),  I = 1, NNODES ),
     .                                  ( Y(I),     I = 1, NNODES )
            END DO
         END IF
        DO I=1,NNODES
          convert absolute (Pa) pressures to gage (KPa)
          P(I) = (P(I)-PAMBS)*1.E-3
          convert absolute (K) temperatures to (C)
          T(I) = T(I)-TZERK
        END DO
        CHECK FOR SI CONVERSION
        IF ( KFSI.NE.1 ) THEN
          DO I = 1, NNODES
            T(I) = T(I) * CVTEMP + 32.0     ! convert (C)  to (F)
            P(I) = P(I) * 1.0E+3 / PTOSI    ! convert (Pa) to (psi) gage
          END DO
```

```
      DO I = 1, NBRCH
        Q(I) = Q(I) * XQSI
      END DO
    END IF

      NOW, EXTRACT THE NODES AND BRANCHES DESIRED FOR PLOTTING
      PURPOSES. A MAXMIMUM NUMBER OF "MCURVE" NODES OR BRANCHES
      MAY BE DISPLAYED.

      KOUNT = KOUNT + 1
      TIMVEC(KOUNT) = TIME
      DO NN = 1, NNPLT
        YFPRES(KOUNT,NN) = P(NNPLTP(NN))          ! NODE PRESSURE
        YFTEMP(KOUNT,NN) = T(NNPLTP(NN))          ! NODE TEMPERATURE
      END DO
      DO NB = 1, NBPLT
        YFVFLO(KOUNT,NB) = Q(NBPLTP(NB))          ! VOLUME FLOW
        YFMFLO(KOUNT,NB) = MFLOW(NBPLTP(NB))      ! MASS FLOW
        YFTHTA(KOUNT,NB) = THETA(NBPLTP(NB))      ! BLADE ANGLES
      END DO

      THIS VERSION OF THE FIRAC POST PROCESSOR DOES NOT ALLOW
      PARTICULATE PRESENTATION.

    GO BACK FOR THE NEXT TIME STEP FROM THE SCRATCH FILE

   END DO
40 CONTINUE

   We have reached the end of the scratch file, either by
   end of record or by end of do-loop.

   NTIME = KOUNT
   WRITE(*,*) ' Number of times read from FIRAC: .... ',NTIME
   WRITE(*,*) ' Initial time read from FIRAC: ....... ',TIMVEC(1)
   WRITE(*,*) ' Final time read from FIRAC: ......... ',TIMVEC(NTIME)

   ------------------------------------------------------------------

   CLOSE(UNIT=ISCR)
   RETURN
   END
```

PC Fire Research

Mechanical Engineering Department – New Mexico State University
P. O. Box 30001, Dept. 3450, Las Cruces, NM 88003
TEL: (505) 646-3501 FAX: (505) 646-6111

FAX COVER SHEET

DATE: 5 Nov 91

TO: Bill Gregory
Los Alamos

FAX NO: 665-2897

FROM: Ed Hensel
NMSU

PAGES FOLLOWING: 3

RE: Info on summary file. This is a list of files that were deleted. A Similar list is being compiled with citation information for

MESSAGE: the abstracts retained in the data base.

Ed

Title: Reliability of Portable Dry Chemical Fire Extinguishers in Use on Motorboats; Final rept.
Title: Waking Effectiveness of Household Smoke and Fire Detection Device Final rept.
Title: Development of Phosphorylated Adhesives
Title: Improved Mine Stench Fire Warning System; NTIS Tech Note
Title: Monoammonium Phosphate: Effect on Flammability of Excelsior and P Needles; Forest Service research paper
Title: Characterization of Borosilicate Glass Containing Savannah River Radioactive Waste. II. Microstructure and Durability
Title: Equivalency Evaluation of Firefighting Agents and Minimum Require at U.S. Air Force Airfields; Final rept.
Title: MIT-GRI (Massachusetts Institute of Technology-Gas Research Insti LNG Safety and Research Workshop. Volume III. LNG Fires - Combust Radiation
Title: Vibration Testing of Railroad Tank Car Specimens; Final rept. Dec Apr 81
Title: Problems Encountered in Welding Inconel 625 Bridgewire Material
Title: The Fabrication and Testing of Glass Matrix Composite Cylinders f Gun Barrel Liner Applications; Final rept. 15 Jan 81-15 Jan 82
Title: Effect of Ventilation on the Rates of Heat, Smoke, and Carbon Mon Production in a Typical Jail Cell Fire
Title: Flame Retardant Additives for Thermoplastics and Thermosets. 1973 1982 (Citations from the Rubber and Plastics Research Association Rept. for 1973-May 82
Title: National Voluntary Laboratory Accreditation Program Proficiency Testing for Thermal Insulation Materials Laboratory Accreditation 9 - August 1983; Final rept.
Title: Park Service Room Fire Test Simulations Using the Harvard Level 5 Computer Fire Model (hotel rooms)
Title: Effect of Hepatotoxic Nonadecafluorodecanoic Acid on Gene Express in Rat Liver; Final rept. 15 Apr 83-14 Jan 84
Title: Acute Inhalation Toxicological Evaluation of Combustion Products Fire Retarded and Non-Fire Retarded Flexible Polyurethane Foam an Final rept.
Title: Combustibility of Electrical Wire and Cable for Rail Rapid Transi
Title: Air Transportable AS32/P-4 Crash Rescue Vehicle; Final rept. Sep May 80
Title: Lightweight Fire Fighting Module Evaluation; Final rept. (Coast G
Title: Plate Thermometer: A Simple Instrument for Reaching Harmonized Fi Resistance Tests. NORDTEST Project 609-86; Final rept
Title: Brandrisker vid Sprutmalning-Hoegtrycksprutning Betydligt Brandfarligare aen Konventionell Sprutmalning (Fire Hazards of Sp
Title: Toxic and Radiological Risk Equivalence Approach in UF6 Transport
Title: Thermal Behavior of the Type 30 B Cylinder Equipped with the 21 P Overpack and Study of Protective Covers for the 48 Y Cylinder Val
Title: Burning of Small Crib Ignition Sources; Research rept.
Title: Evaluation of Forest Fire Burn Model of Reaction Kinetics of Heterogeneous Explosives; Summary rept. 1987-1988
Title: Bulldozer Fireline Production Rates-1988 Update; Forest Service research paper
Title: Products of Wood Smolder and Their Relation to Wood-Burning Stove
Title: Flame Roll-Out Study for Gas Fired Water Heaters
Title: Jeffcott Equations in Nonlinear Rotordynamics
Title: Effects of Additives on Fire Properties of Polyethylene; Technica memo
Title: Fire Retarding Coatings. January 1980-December 1987 (Citations fr World Surface Coatings Abstracts); Rept. for Jan 80-Dec 87
Title: Flame Retardant Polymeric Foams: Manufacturing, Applications, and Hazards. July 1984-August 1987 (Citations from the Rubber and Pla

re Detection System. Volume
Title: Creosote Accumulation as a Function of Moisture Content, Wood Spe
Title: Ferrous Metal Fires; Final rept (Marine app)
Title: Coyote Series for 40-M exp 3 Liquefied Natural Gas (LNG) Dispersi RPT, and Vapor Burn Tests
Title: Research on Fire-Resistant Diesel Fuel; Interim rept. 1 Oct 79-31 81
Title: PAT-2 (Plutonium Air-Transportable Model 2) Safety Analysis Repor
Title: Ships Lounge Burnout Experiments; Final rept.
Title: Fast Repsonse Sprinklers in Patient Room Fires; Final rept.
Title: Numerical Experiments on the Shock Sensitivity of Munitions; Fina
Title: Smoke Movement and Smoke Control on Merchant Ships; Final rept.
Title: Thermal Characteristics of Thick Red Oak Flakeboard; Forest Servi research paper
Title: Reduced-Scale Modeling of Mobile Home Fires: A Progress Report; Interim rept.
Title: Fire Endurance Under Design Load for Walls of One-Story Three-Roo
Title: Investigation of Solar Collector Fire Incident
Title: Technical Support for the Consumer Product Safety Commission 1979 Interim Standards for Cellulose Insulation; Final rept.
Title: Report on 1980 Property Loss Comparison Fires: An Insurance Indus Assessment of the Impact of Residential Sprinklers on Fire Loss
Title: Evaluation of Alternate Mounting Methods for the Evaluation of Brattice Cloth on ASTM E-162; Final rept.
Title: Toxicity of Smoke during Chair Smoldering Tests and Small Scale T Using the Same Materials; Final rept.
Title: Determination of Ammunition Training Rates for Marine Forces Stud Volume 1; Final rept.
Title: Modeling of NBS (National Bureau of Standards) Mattress Tests wit Harvard Mark V Fire Simulation; Final rept.
Title: COFT (Conduct of Fire Trainer) - A New Concept in Tank Gunnery Training
Title: Changing Artillery Training Requirements
Title: Prevention of Spontaneous Combustion in Coal in Stockpiles in Sou Africa
Title: Control Technology Assessment for Coal Gasification and Liquefact Processes, Exxon Coal Liquefaction Pilot Plant (ECLP), Baytown, T Final rept.
Title: Intensity and Duration of Chimney Fires in Several Chimneys
Title: Fire Research on Seismically Damaged Concrete Beams Repaired with Epoxy Adhesives; Final rept.
Title: Abbreviated Independent Evaluation Report on the Survey Electroni Distance Measuring Equipment - Medium Range (SEDME-MR); Final rep 83
Title: Provning av Explosionssaekra Kaerl (Testing of Explosion-Proof Containers)
Title: Investigation of the Geokinetics Horizontal in-Situ Oil-Shale Retorting Process. Quarterly Report, April-June 1983
Title: Fire-Resistant Hydraulic Fluids and Lubricants: Phosphzene-based materials are evaluated; NTIS Tech Note
Title: Laboratory Fire-Side Corrosion Evaluation of Improved Superheater Alloys and Coatings. Final Report (With Addition of Aluminium, Si Manganese)

Association Database); Rept. for Jul 84-Aug 87

Title: Powder Extinguishants for Jet-Fuel Fires: Mixtures of alkali meta
dawsonite and metal halide show superior performance; NTIS Tech N
Performing Organization: National Aeronautics and Space Administr
Washington, DC.

Title: Fire Retarding Coatings. 1980-1986 (Citations from World Surface
Coatings Abstracts); Rept. for 1980-86

Mechanical Engineering Department – New Mexico State University
P. O. Box 30001, Dept. 3450, Las Cruces, NM 88003
TEL: (505) 646-3501 FAX: (505) 646-6111

FAX COVER SHEET

DATE: 14 April 1992

TO: Stan Claybrook FAX NO: 509-943-6617

FROM: Ed Hensel

PAGES FOLLOWING: 5

RE: Copy of the email I sent to Bill Gregory on Sunday, describing typical errors in FIRAC.

MESSAGE:

QUESTIONS COMMENTS AND CHANGES BY Ed Hensel

history.txt - documentation file for making sure that Hensel and Gregory remain consistent while working long distance.

```
******************************************************************************
*** 11 April 1992:
******************************************************************************
```

I was under the impression that Stan Claybrook had taken care of the memory mapping errors. This is not correct. There still exist SEVERE memory mapping errors within FIRAC. As soon as I run the code with /check=all on my vaxstation, we once again have IDENTICALLY the same errors as before. I did a "difference" comparison of the old code (firacn2) and the new code (firacn2s) from Stan. There were very few changes from the old version I had. Several arrays (IDSPEC for example) are not dimensioned consistently throughout the code. The following is a summary of what I encounter:

```
  >>>>>>>>>>>>>>>>>>>>>>  start of DEBUG Session  <<<<<<<<<<<<<<<<<<<<<<<<<<<
        VAX DEBUG Version V5.5-021

%DEBUG-I-INITIAL, language is FORTRAN, module set to MAIN
%DEBUG-I-NOTATMAIN, type GO to get to start of main program

DBG> go
break at routine MAIN
    76:         CALL BDATA
DBG> go
********** FIRAC **********

Input file name?
prob_1.inp
Output file name?
prob_2.inp

Processing . . .
%DEBUG-I-DYNMODSET, setting module RPDATA
%SYSTEM-F-SUBRNG, arithmetic trap, subscript out of range at PC=00201022, PSL=03
C00060
break on unhandled exception preceding RPDATA\%LINE 26+176
    26:         READ(IREAD,200) ISPEC(IS),IDSPEC(IS,1),IDSPEC(IS,2),DIAP(IS),
DBG>set scop rpdata
DBG> sh calls
 module name     routine name                        line       rel PC    abs PC
*RPDATA          RPDATA                                26     00000102  00201022
 INPROC                                                       00000000  001F5D52
*MAIN            MAIN                                  77     00000010  001EEE10
                 FOR$INIT_UNDER                               00000925  00213425
                 FOR$INIT_UNDER                               0000000B  00212B0B
                 FOR$INIT_UNDER                               00000900  00213400
DBG> type 10:27
module RPDATA
    10:         COMMON /PARPT/ DFLUX(100,100),PMOF(100,100),Y(100,100),
    11:        1 WMASS(100,100),SRCE(100,100),SINK(100,100)
    12:         COMMON /PARVLT/ DIAP(100), RHOP(100), ISPEC(100), IDSPEC(100,2)
    13:         INTEGER MSG(5), IDSPEC
    14:         COMMON/LOGNRM/DG(5),SIGMA(5)
    15:         DIMENSION DP(20),DUM(5),DMAX(5),DMIN(5),YMIN(5),NBINS(5),
    16:        1 TEMP(100,5)
    17:         DATA MSG /4H CAL,4HL FR,4HOM R,4HPDAT,1HA/
    18:         DATA RZERO /0.0/
    19: C
```

```
    20:         IF (NSPECES.LE.0) GOTO 160
    21:         READ (IREAD,180) DUMMY
    22:         DO 140 IS=1,NSPECES
    23: C
    24: C           READ PARTICULATE SPECIES DATA CARDS
    25: C
    26:         READ(IREAD,200) ISPEC(IS),IDSPEC(IS,1),IDSPEC(IS,2),DIAP(IS),
    27:        1 RHOP(IS),SIGMA(IS),YMIN(IS)
DBG>ex IS,NSPECES,SIGMA,YMIN
RPDATA\IS:       6
RPDATA\NSPECES: 13
RPDATA\SIGMA
    (1):         0.0000000000000000
    (2):         0.0000000000000000
    (3):         0.0000000000000000
    (4):         0.0000000000000000
    (5):         0.0000000000000000
RPDATA\YMIN
    (1):         0.0000000000000000
    (2):         0.0000000000000000
    (3):         0.0000000000000000
    (4):         0.0000000000000000
    (5):         0.0000000000000000
DBG>
  >>>>>>>>>>>>>>>>>>>>>>  end of DEBUG Session  <<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

Clearly, the code immediately accesses array elements which do not exist. This can not be correct. In addition to this problem, IDSPEC is STILL not properly dimensioned -- it is different in common GASVLT.

I went ahead and ran the code with \CHECK=NONE, to see what would happen. I got a floating point overflow, just like I had before. Here is the error message:

```
   (several times printed out to the screen)
     :
     :
     :
   TIME (SEC) =    1.995000000000002
   TIME (SEC) =    1.996000000000002
   TIME (SEC) =    1.997000000000002
   TIME (SEC) =    1.998000000000002
   TIME (SEC) =    1.999000000000002
   %SYSTEM-F-FLTOVF_F, arithmetic fault, floating overflow at PC=0020B0F1, PSL=0
   0000
   %TRACE-F-TRACEBACK, symbolic stack dump follows
   module name     routine name                     line      rel PC    abs PC

   DEPLETE         DEPLETE                            34    00000249  0020B0F1
   PDMECH          PDMECH                             97    000007E4  0020A4DC
   FIRIN           FIRIN                            1254    00005280  00207BF0
   TFNS            TFNS                              107    00000039  001FE8D1
   SOLVE           SOLVE                              96    0000004C  001FE36C
   MAIN            MAIN                               79    0000001E  001EEC1E
```

From what I can see of the code, it does not look as though the errors I pointed out earlier have been corrected.

The manner in which the 3-d array QRAD2 is manipulated appears to be incorrect. For example, in the FIRAC version of FIRIN, we have the code sequence:

```
      DO 1120 I=1,9
         :
         :
C Identify contaminated combustibles for radioactive release calculation
```

```
          IF (NRAD(1).EQ.0 .AND. NRAD(2).EQ.0)GO TO 1115
           DO 1110 JACT=1,10
            DO 1111 IFORM=1,3
             NJ=((IFORM*10)-10)+I
             QRAD1(NJ,JACT,1)=QRAD1(NJ,JACT,2)+QRAD1(NJ,JACT,1)
 1111 CONTINUE
            DO 1110 JFORM=1,2
C             HENSEL 11 APRIL 92. I     VARIES FROM  1 TO 9,
C                                 JFORM VARIES FROM  1 TO 2.
C                       THEREFORE, NJ    VARIES FROM -5 TO 6.
C             THIS MUST BE AN ERROR, AND THE SEQUENCING APPEARS VERY
C             SUSPECT.
              NJ=(I-6)+(JFORM-1)*3
              QRAD2(NJ,JACT,1)=QRAD2(NJ,JACT,2)+QRAD2(NJ,JACT,1)
 1110       CONTINUE
 1115     ENDIF
      TAREC=TAREC+AREC(I,IBO)
 1120 CONTINUE
```

The reference to negative subscripts in the QRAD2 array will obviously cause difficulties. This same potential error also exists in the code FIRIN PC, although I have not yet encountered it a run time in FIRIN PC. Elsewhere in the code, the I loop varies from 7 to 9, which would cause NJ to vary from 1 to 6, which is at least reasonable. In subroutine INPROC, we encounter another error with sample problem 1:

```
  >>>>>>>>>>>>>>>>>>>>>  start of DEBUG Session  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<
DBG> TY 970:980
module INPROC
   970:  2541 IF(NRAD(J).EQ.0)GOTO 2531
   971:       NLAST=NRAD(J)
   972:       DO 2551 K=1,NLAST
   973:       IF(J.GT.2)GOTO 2561
   974:       READ(IREAD,2620) IFORM,I,JACT,IBO
   975:       NJ=((IFORM*10)-10)+I
   976:       IF(J.EQ.1)READ(IREAD,2700)QRAD1(NJ,JACT,IBO)
   977:       IF(J.EQ.1)QRAD1(NJ,JACT,IBO)=QRAD1(NJ,JACT,IBO)*453.6
   978:       IF(J.EQ.2)READ(IREAD,2700)QRAD2(NJ,JACT,IBO)
   979:       IF(J.EQ.2)QRAD2(NJ,JACT,IBO)=QRAD2(NJ,JACT,IBO)*453.6
   980:       GOTO 2551
DBG> ex IFORM,I,JACT,IBO
INPROC\IFORM:    1
INPROC\I:        7
INPROC\JACT:     1
INPROC\IBO:      2
DBG> ex QRAD2(NJ,JACT,IBO)
%DEBUG-I-SUBOUTBND, subscript 1 is out of bounds
-DEBUG-I-VALRNG, value is 7, bounds are 1..6
INPROC\QRAD2(7,1,2):    0.0000000000000000
DBG>
  >>>>>>>>>>>>>>>>>>>>>  end of DEBUG Session  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

? compare against old version

Clearly, this is a FIRAC code error. The inputs read from the data file are well within the bounds of those specified in the user manual. These errors will certainly be encountered at some point be evaluators as they use the code on a variety of problems.

Bill, I must register some very serious concern at this point. I do not see how all of these errors can possibly be fixed and the code can be even reasonably validated by May 15, let alone the addition of new capabilities. I believe it would be a mistake to release a fire simulation program that has known and identified serious flaws. I was under the impression that more than one person was working on the code in Washington to correct these numerous errors. Yet, when I look closely at the code which has been returned, it is obvious that very few modifications have been made to the code. This program simply can not be correct.

The fact that the variable IDSPEC is used for both gaseous and particulate species will cause a difficult problem if a user tries to run a problem with both types present in the scenario.

I don't see any way we are going to get a RELIABLE version of FIRAC with particulate model running by May. Even if we get numbers out of the code, I don't trust them, based on the kinds of errors I am seeing. I need some guidance on what I should do for the next several weeks. I do not feel comfortable about just using the code as it runs under SVS fortran, because I know that there are errors in the code. I fully believe that these errors will come back to haunt us if this code is released to a group of DOE evaluators.

```
************************************************************************
*** 9 April 1992:
************************************************************************
```

Received new SVS fortran version of FIRACN2S.F source code from Stan Claybrook, along with an input deck for problem number 1.

```
************************************************************************
*** 20 February 1992:
************************************************************************
```

I have encountered what I believe to be a fairly pervasive error throughout firac. The error is the same in in all versions of FIRAC I have seen so far, including the most recent version FIRACN2.FOR. The dimension of arrays referring to species concentrations are not consistenly dimensioned to the right size. The problem is best illustrated by sample problem 1, with the number of NGSPECE=13. In routine RGDATA (called from INPROC) the code attempts to read

```
          READ (IREAD,200) ISPEC(IS),IDSPEC(IS,1),IDSPEC(IS,2)
```

with the value of IS = 10. However, the arrays are only dimensioned to a size of

```
          COMMON /GASVLT/ ISPEC(5), IDSPEC(5,2)
```

In addition, RGDATA calls CLEAR2 to zero out columns of arrays in

```
          COMMON /GASPT/ GFLUX(100,5),GFLUXI(100,5),GCON(100,5)
```

which do not even exist! This results in memory mapping errors which call into serious question the validity of the results obtained with the code. Indeed, when I compile the code with full debugging options implemented on the VMS compiler, it becomes clear that array access violations are common throughout the code. I feel that these errors are critical issues that must be resolved before the output from the FIRAC code can be trusted. The errors are memory mapping errors of a nature that may very well not be detected by many FORTRAN compilers. I believe that it is essential to perform an in-depth verification of the code as soon as possible. It appears that many changes have been implemented in the code over a period of time, and the code may not have been fully verified following completion of many such changes. Of particular concern are subtle errors associated with changing the dimensions of arrays and problem sizes.

```
************************************************************************
*** 2 February 1992:
************************************************************************
```

(1) Crunched program to remove line numbers and trailing spaces. Reduced the source code file size to less than one-half of original size.
(2) Modified several FORMAT statements to remove Hollerith fields.
(3) Commented out SUN specific calls for time and date.
(4) Made changes to SUBROUTINE MASSGE, FORMAT 1900 and 1920. Changed WIDTH(I) to WIDTH(K), added 'Width' to column headers, and changed the output from 4 columns wide to 3 columns wide to

avoid output record overflow.
(5) Added open statments for I/O files.
(6) Changed logical unit numbering to avoid future conflicts with FIRINPC and graphics codes. Rather than using an output file called FOUT.DAT, the tabular output will be written to a file called SUMRY.DAT, consistent with the FIRIN PC code.
(7) Changed the definition of a small machine number from 10**-64 to 10**-37 for more common fortran compiler limits on 32 bit machines.
(8) The conversion factors PTOSI, CVTEMP, QTOSI did not appear to be properly initialized upon entry to SUBROUTINE MASSGE, causing an underflow error. Upon debugging, it became obvious that the CALL CLEAR statements in BDATA wre wiping out the data initialization of the conversion factors. It appears that SUBROUTINE CLEAR has a bug, since the integers passed to it are typically 4 byte integers, while the (a-h,o-z) values are 8 byte values. This causes an alignment problem when the integer values are improperly cleared. Further, it appears that the number of elements cleared may not be consistent with the declared lengths. Several vectors were declared to length 100, but cleared only to 50. After these changes, the data type conversion information appeared to be passed to MASSGE properly.
(9) Some of the output data contained in the sample output file sent to Hensel from Gregory appeared as thought there may have been an memory allocation error in the SUN version. FOr example, in the version revised by Hensel, the output file contains:

```
************
************
File $1$DKA200:[EHENSEL.FIRAC]SUMRY.DAT;1
 2260      VOLUME      0.00000E+00  0.00000E+00  0.00000E+00  0.00000E+00  0.0
 2261                    0.00000E+00  0.00000E+00
 2262
 2263    TOTAL SPECIE AIRBORN VOLUME (M**3) ; 0.000E+00
 2264
```

whereas the file provided by Bill Gregory contains

```
******
File $1$DKA200:[EHENSEL.FIRAC]BILL2.DAT;1
 2272      VOLUME__      4.94066-324  0.00000E+00  0.00000E+00  0.00000E+00  0.0
 2273                    0.00000E+00  4.94066-324
 2274
 2275    TOTAL SPECIE AIRBORN VOLUME (M**3) ; 9.881-324
 2276
************
************
```

This must be tracked down to verify the cause of the difference between the two versions of the code.

(10) What is logical unit number 29 used for? Can these references be deleted? It looks like it may have been used for debugging at one point in time.

(11) Can logical unit nmber 17 (IPLOT) be removed?

(12) I had difficulty reading the wordperfect file you mailed me. It looks as though it got garbled in transmission, since it was a long record length binary file. We may have to send this file by hard floppy.

FIRAC Code Support

Edward Hensel, Associate Professor
Principal Investigator
Mechanical Engineering, Department 3450
New Mexico State University
Las Cruces, New Mexico 88003-0001
Phone (505)646-2011 FAX (505)646-6111

October 19, 1992

1 Executive Summary

FIRAC has grown into a comprehensive ventilation system model over the years. In contrast with several other popular codes now available, FIRAC is designed to model a complex network of interconnected ducts, pipes, gloveboxes, filtration devices, and flow control devices. FIRAC currently contains a relatively simplistic fire compartment model, based on the FIRIN computer code. One appealing feature of FIRAC is its amenability to modularity – additional (or replacement) fire compartment modules can be integrated into the FIRAC system response code. In the discussion that follows, we will address the *total system* capabilities of the FIRAC code and recommend tasks that will make FIRAC a fully supported and modular code useful for fire protection engineers throughout the DOE community. In the effort that follows, we will assume that another team is jointly working on continued development of a fire compartment module (such as CFAST or FIRIN) that can be coupled with the complete fire system library.

A summary flow diagram of the FIRAC code is presented in Figure 1. This diagram clearly shows the manner in which FIRAC may be coupled to various fire compartment modules. Basically, the fire compartment module is used to determine the localized effects of a fire within a room or an interconnected series of rooms. CFAST is a popular NIST code capable of modeling a ventilated fire in a series of a few rooms. FIRIN is a simplistic two-layer model which models a fire in a single compartment. Other fire models may in the future be proposed, for example, to perform a complete finite difference model of the fire compartment to study the effects of natural and forced convection, sprinkler systems, etc. The unique capabilities of FIRAC come into play when we examine the rest of the system, not in direct contact with the fire compartment. FIRAC calls the compartment module at each increment of time to determine the particulate and heat generation by the fire, along with the corresponding disturbance to the mass flow. FIRAC treats the fire compartment as a "boundary" node in the system model, and also allows other boundary nodes, such as outside air intakes and exhausts,

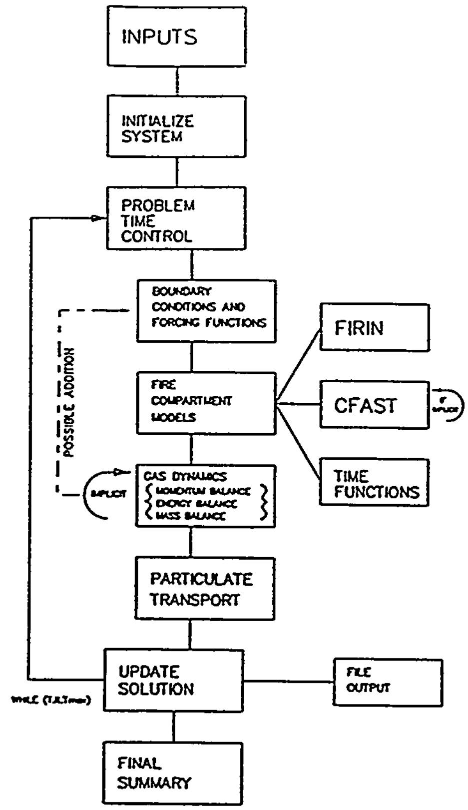


Figure 1: FIRAC flow diagram.

blowers, dampers, etc. FIRAC uses the boundary information supplied by user functions and the fire compartment to perform a transient analysis of the full ventilation system.

Following the momentum and energy balance on the entire ventilation system, FIRAC computes a particulate distribution and deposition throughout the system, including the effects of filtration. The unique capabilities of FIRAC include plutonium transport and radionuclide models, based on empirical data. While such particulate models are of utmost importance within the DOE complex, due to criticality and health physics concerns, such particulate transport capabilities are not present in most fire compartment codes such as CFAST. Indeed, it may well be argued that it is undesireable for widely available codes such as CFAST to incorporate such capabilities.

The work proposed here will modernize the FIRAC code, using software and hardware technology that simply was not available at the time FIRAC was originally written. In addition, the increased modularity of the new FIRAC will allow it to be coupled to more sophisticated fire compartment modules such as FIRIN. All of the work will be carried out with a rigorous set of validation and verification tests at each stage to insure the integrity of the model throughout the process. The work is presented as a sequence of five tasks, with deliverables identified at each step, to insure timely completion of the overall project.

2 Task Description

FIRAC was originally developed on large CRAY-class supercomputers and mainframes with severe memory limitations. As computer technology has advanced, the capabilities of FIRAC have been increased to allow larger models, more particulate species, various materials, etc. However, not all of the changes have been accompanied by a full suite of verification tests at each step. A number of memory allocation and array mapping errors have been identified and corrected in the current version of the code. We feel it is necessary to review FIRAC for such errors, and to revise the code to guard against the future introduction of such errors.

Rather than having CRAY, SUN, and PC versions of the code, we will have a *single* set of FORTRAN source modules that can be recompiled and linked together on various machines. This would eliminate our current problem of one version apparently running successfully, while another version can not. More powerful computers will have access to all of the code modules, while personal computers will have access only to those modules that can be realistically executed on a given machine. For example, a fire compartment module based on FIRIN may be appropriate to run on a PC or workstation. A fire compartment model with full sprinkler support may require a 3-d transient simulation, and could only be run on a large supercomputer.

Task 1 The code will be brought up to current software standards. Much of the FIRAC code is written on the FORTRAN 66 and FORTRAN V standard. FIRAC will brought to the FORTRAN 77 standard, and will be fully modularized. As more FORTRAN 90 compilers enter the market, the new code will be much easier to maintain and can be moved to the FORTRAN 90 and subsequent standards in a modular fashion. The deliverable at the end of Task 1 will be a modern FIRAC that is amenable to steady and continued software maintenance. The capabilities of FIRAC will be essentially the same as the current code, but FIRAC will now be a code using the capabilities of modern computer technology. Specifically, we will accomplish the following.

1. All hard-wired array dimensions will be removed from declaration statements, and replaced with Fortran PARAMETERS. This single step will dramatically enhance code maintainability, since it will now be clear when an array dimension refers to the number of rooms, branches, species, etc.
2. All common blocks be removed from the code, in favour of argument lists. This will allow us to identify all appropriate code dependencies, and to establish a formal program verification methodology. We can then fully debug one module at a time, without worrying that we introduce an error in an auxiliary code segment.
3. All variables within each module will be identified as either INPUT, OUTPUT, LOCAL or PARAMETER variables. The declaration characteristics of each variable (*e.g.*, REAL*4) will be identified. The engineering units of each variable (*e.g.*, m^3/sec) will be given in the argument dictionary to insure proper calling syntax and ease future maintenance.
4. The energy and momentum balance portions of the code will be fully validated.

5. A programmer's reference manual will be written that describes the theory and implementation of each subruotine and function. Each module will have a validation test case provided that will fully exercise the capabilities of the module. The validation cases will be used by later programmers to insure reliablity of the code as it undergoes future maintenance and updates.

Task 2 FIRAC will rewritten for more efficient code execution. The input and output processing capabilities of the code will be brought to modern standards. A tighter coupling between the currently available pre- and post-processors will be attained.

1. The execution times and memory requirements of the code will be profiled. Modern CASE (Computer Aided Software Engineering) tools will be used to determine which modules require the most resources. We believe we will learn how to properly combine certain utility subroutines to improve runtime performance by limiting stack pops.
2. Modules INPROC and OUTPRC will be improved. Extraneous output, a holdover from the days of huge line printer files, will be eliminated.
3. The particulate transport modules will be studied and validated. Recent particulate trasport theory developments and empirical data will be incorporated. Much of this work was not avaliable when FIRAC was first written. This step will illustrate the value of modularity for code maintenance to insure that the simulation remains current with empirical and theoretical progress.
4. The programmer's reference manual will updated to reflect the newest changes to the code.
5. The working code at the end of this task will be more efficient, more modular, and will conatin the most recent particulate models available.

Task 3 FIRAC will be coupled with additional fire compartment modules. We will retain the time function capabilities of FIRAC (for use on small PCs, etc.) and hope to retain the FIRIN compartment module, if appropriate. We will employ the NIST code CFAST as a more sophisticated fire compartment module. The argument lists and communcation between the two codes, FIRAC and CFAST, will be studied in detail. We will work closely with collaborators at LANL and NIST to insure that this phase of the project proceeds on schedule.

1. We will obtain a detailed input and output argument list list from CFAST. We will write an interface subroutine to perform all required units conversion as well as mass, energy, and momentum balances between the FIRAC ventilation system code and the CFAST fire compartment code.
2. We will retain CFAST as a separate library, so it can be run as a stand-alone code (as NIST currently runs it) or as a callable subroutine from within FIRAC. NIST will be responsible for providing such a callable library and a stand-alone driver.
3. Substantial effort will be directed towards validation. We will run all NIST provided standard test cases on the standalone NIST code, and on the code as called from FIRAC, to insure that we have not introduced any discrepancies.
4. We will demonstrate that the combined power of the two codes dramatically exceeds the capability of either code when running stand-alone.

5. The programmer's reference manual will updated to reflect the newest changes to the code.
6. A user's guide will be started.

Task 4 The key issue here will be to study the effects of interaction between the codes. We will investigate what FIRAC will need to supply each compartment model to ensure reliable exection.

1. We will further exercise the codes and their interface. We will develop test cases for a variety of scenarios in order to validate the physics of the code and to make sure that the two codes are interchanging information properly.
2. We will release a preliminary version of the combined codes to a select group of users. These users will undoubtedly discover bugs and request features that are not available. We will work with the users to resolve issues as they arise. First and foremost will be to solve any bugs.
3. The PhD student will place increased emphasis on the particulate transport module and its validation.
4. Numerous simulation test cases will be executed. This will include analysis of any test cases that present problems to the select user group.
5. The programmer's reference manual will updated to reflect the newest changes to the code.
6. The user's guide will be updated, relying upon feedback from the core user group.

Task 5 Experimental verification of the code will proceed. We will rely on existing experimental data from sites around the world.

1. The programmer's reference manual will updated. A suite of benchmark test cases will be included for subsequent code maintenance and validation.
2. The user's guide will be updated. This final version will include directions for installation of the code on a variety of machines, soft copy of all sample problems, and an error dictionary.
3. FIRAC, the user's guide, and the programmer's reference manual will be released for production usage.
4. The PhD student will present suggestions for future enhancements of the particulate transport modules.

Budget

Item	Description Task Duration	Task 1 5 Mos	Task 2 5 Mos	Task 3 5 Mos	Task 4 5 Mos	Task 5 5 Mos
1.	Dr. Edward Hensel 20 @ $280/day 26.0% Benefits	 5,600 1,456	 5,600 1,456	 5,600 1,456	 5,600 1,456	 5,600 1,456
2.	MS Student Engineer @ $1,000/month 1.5% Benefits	 5,000 75	 5,000 75	 5,000 75	 5,000 75	 5,000 75
3.	MS Student Engineer @ $1,000/month 1.5% Benefits	 5,000 75	 5,000 75	 5,000 75	 5,000 75	 5,000 75
4.	PhD Student Engineer @ $1,200/month 16.0% Benefits	 6,000 960	 6,000 960	 6,000 960	 6,000 960	 6,000 960
5.	Undergrad Student @ $500/month 1.5% Benefits	 2,500 38	 2,500 38	 2,500 38	 2,500 38	 2,500 38
6.	Computer SYSADMIN @ $100/month 26.0% Benefits	 500 130	 500 130	 500 130	 500 130	 500 130
7.	Travel to LANL @ $250/trip	 250	 250	 250	 250	 250
8.	Supplies Photoreproduction Telecommunications Shipping/Express Mail	150 50 100 50	150 50 100 50	150 50 100 50	150 50 100 50	150 50 100 50
9.	Subtotal Direct Costs	27,934	27,934	27,934	27,934	27,934
10.	Indirect Cost 44.7% Direct Costs	 12,486	 12,486	 12,486	 12,486	 12,486
11.	**TOTAL TASK COST**	40,420	40,420	40,420	40,420	40,420

Investigator Qualifications

EDWARD HENSEL, Ph.D., PE

Dr. Hensel is an internationally recognized authority in the field of inverse heat conduction analysis. He recently authored a textbook titled "Inverse Theory and Applications for Engineers" (Prentice Hall, 1991). Additionally, he has jointly authored several papers related to parallel computing for engineering applications.

INVITED LECTURES

Two-dimensional Nonlinear Inverse Heat Conduction Problem Case Study, International Conference on Identification in Dynamic Systems, Suzdal, USSR, Sept. 10-14, 1990.

Sensor Placement Considerations for Multi-dimensional Inverse Problems, International Conference on Finite Elements in Flow Problems, Huntsville, AL, April 3–7, 1989.

Inverse Problems for Multi-dimensional Parabolic Partial Differential Equations, International Conference on Inverse Design Concepts and Optimization in Engineering Sciences - II, October 26-28, 1987.

SELECTED ARTICLES

Hensel, E., and Dalton, K., *Nonintrusive Characterization of Waste Sites,* Spectrum '92, American Nuclear Society, Boise, ID, August 1992.

Dalton, K. and Hensel, E., *Feasability Study of Inverting Near-Field Electromagnetic Scattering Data for Subsurface Characterization,* ECOWorld Conference, Washington DC, June 1992.

Hutchinson, S., Hensel, E., Castillo, S., Dalton, K., *The Finite Element Solution of Elliptical Systems on a Data Parallel Computer*, International Journal for Numerical Methods in Engineering, Vol. 32, pp. 347–362, 1991.

Hensel, E., Hills, R., *Steady State Two-Dimensional Inverse Heat Conduction*, Numerical Heat Transfer, Part B, Vol 15, pp. 227-240, 1989.

Hensel, E., *Inverse Problems for Multi-dimensional Parabolic Partial Differential Equations*, Invited paper for Applied Mechanics Reviews, Vol. 41, No. 6, June 1988, ASME Book No. AMR038.

Hills, R., Raynaud, M., Hensel, E., *Surface Variance Estimates Using an Adjoint Formulation for a One Dimensional Nonlinear Inverse Heat Conduction Technique*, Numerical Heat Transfer, Vol. 10, pp. 441-461, 1986.

Hensel, E., Hills, R.G., *An Initial Value Approach to the Inverse Heat Conduction Problem*, ASME Journal of Heat Transfer, Vol. 108, No. 2, May 1986, pp. 248–256.