

NIST Special Publication 1017-1

User's Guide for Smokeview Version 5  
- A Tool for Visualizing Fire Dynamics  
Simulation Data  
(Draft: May 28, 2007)

Glenn P. Forney

NIST Special Publication 1017-1

# User's Guide for Smokeview Version 5 - A Tool for Visualizing Fire Dynamics Simulation Data

Glenn P. Forney  
*Fire Research Division  
Building and Fire Research Laboratory*

June 2007



U.S. Department of Commerce  
*Carlos M. Gutierrez, Secretary*

Technology Administration  
*Robert Cresanti, Under Secretary for Technology*

National Institute of Standards and Technology  
*William A. Jeffrey, Director*

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

**National Institute of Standards and Technology Special Publication 1017-1**  
**Natl. Inst. Stand. Technol. Spec. Publ. 1017-1, 116 pages (June 2007)**  
**CODEN: NSPUE2**

U.S. GOVERNMENT PRINTING OFFICE  
WASHINGTON: 2004

---

For sale by the Superintendent of Documents, U.S. Government Printing Office  
Internet: [bookstore.gpo.gov](http://bookstore.gpo.gov) – Phone: (202) 512-1800 – Fax: (202) 512-2250  
Mail: Stop SSOP, Washington, DC 20402-0001



# Preface

Smokeview is a software tool designed to visualize numerical calculations generated by the NIST Fire Dynamics Simulator (FDS), a computational fluid dynamics (CFD) model of fire-driven fluid flow. For details on setting up and running FDS cases read the FDS User's guide[1]. These two tools are typically used together to respectively simulate and visualize the flow of smoke induced by a fire. Smokeview visualizes the smoke using traditional scientific methods such as displaying tracer particle flow, 2D or 3D shaded contours of gas flow data such as temperature and flow vectors showing flow direction and magnitude. Smokeview also visualizes smoke realistically so that one can *experience* the fire. This is done by displaying a series of partially transparent planes where the transparencies in each plane (at each grid node) are determined from soot densities computed by FDS. Smokeview also visualizes static data at particular times again using 2D or 3D contours of data such as temperature and flow vectors showing flow direction and magnitude.

Windows PC versions of Smokeview and FDS and associated documentation may be downloaded from the web site <http://fire.nist.gov/smokeview> at no cost. Versions for Linux and Mac/OSX may also be downloaded from the same location.



# Disclaimer

The US Department of Commerce makes no warranty, expressed or implied, to users of Smokeview, and accepts no responsibility for its use. Users of Smokeview assume sole responsibility under Federal law for determining the appropriateness of its use in any particular application; for any conclusions drawn from the results of its use; and for any actions taken or not taken as a result of analysis performed using this tools.

Smokeview and the companion program FDS is intended for use only by those competent in the fields of fluid dynamics, thermodynamics, combustion, and heat transfer, and is intended only to supplement the informed judgment of the qualified user. These software packages may or may not have predictive capability when applied to a specific set of factual circumstances. Lack of accurate predictions could lead to erroneous conclusions with regard to fire safety. All results should be evaluated by an informed user.

Throughout this document, the mention of computer hardware or commercial software does not constitute endorsement by NIST, nor does it indicate that the products are necessarily those best suited for the intended purpose.





# Acknowledgements

A number of people have made significant contributions to the development of Smokeview. In trying to acknowledge those that have contributed, we are inevitably going to miss a few people - let us know and we will include those missed in the next version of this guide.

The original version of Smokeview was inspired by Frames, a visualization program written by James Sims for the Silicon Graphics workstation. This software was based on visualization software written by Stuart Cramer for an Evans and Sutherland computer. Frames used tracer particles to visualize smoke flow computed by a pre-cursor to FDS. Judy Devaney made the NIST multi-screen eight foot Rave facility available allowing a stereo version of Smokeview to be built that can display scenes in 3D. Both Steve Satterfield and Tere Griffin on many occasions helped me demonstrate Smokeview cases on the Rave inspiring many people to the possibility of using Smokeview as a *virtual reality-like* fire fighter training facility.

Many conversations with Nelson Bryner, Dave Evans, Anthony Hamins and Doug Walton were most helpful in determining how Smokeview could be adapted for use in fire fighter training applications.

Smokeview would not be possible without the use of a number of software libraries developed by others. Mark Kilgard while at Silicon Graphics developed GLUT, the basic tool kit for interfacing OpenGL with the underlying operating system on multiple computer platforms. Paul Rademacher while a graduate student at the University of North Carolina developed GLUI, the software library for implementing the user friendly dialog boxes.

Significant contributions have been made by those that have used Smokeview to visualize complex cases; cases that are used to perform both applied and basic research. The resulting feedback has improved Smokeview as a result of their interaction with me, pushing the envelope and not accepting the status quo.

For applied research, Daniel Madrzykowski, Doug Walton and Robert Vettori of NIST have used Smokeview to analyze fire incidents. Steve Kerber has used Smokeview to visualize flows resulting from PPV fans. David Stroup has used Smokeview to analyze cases for use in fire fighter training scenarios. Conversations with Doug Walton have been particularly helpful in identifying needed features and clarifying how best to make their implementation user friendly. David Evans, William (Ruddy) Mell and Ronald Rehm used Smokeview to visualize *urban-wildland interface* fires. For basic research, Greg Linteris has used Smokeview to visualize fire simulations involving the cone calorimeter. Anthony Hamins has used Smokeview to visualize the structure of CH<sub>4</sub>/air flames undergoing the transition from normal to micro-gravity conditions and fire suppression in a compartment. Jiann Yang has used Smokeview to visualize smoke or particle number density and saturation ratio of condensable vapor.

This user's guide has improved through the many constructive comments of the reviewers Anthony Hamins, Doug Walton, Ronald Rehm, and David Sheppard. Chuck Bouldin helped port Smokeview to the Macintosh.

Many people have sent in multiple comments and feedback by email, in particular Adrian Brown, Scot Deal, Charlie Fleischmann, Jason Floyd, Simo Hostikka, Bryan Klein, Davy Leroy, Dave McGill, Brian McLaughlin, Derek Nolan, Steven Olenick, Stephen Priddy, Boris Stock, Jason Sutula, Javier Trelles, and Christopher Wood.

Feedback is encouraged and may be sent to [glenn.forney@nist.gov](mailto:glenn.forney@nist.gov).



# Contents

<b>Preface</b>	<b>i</b>
<b>Disclaimer</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Overview . . . . .	1
1.2 Features . . . . .	2
1.3 What's New . . . . .	4
<b>2 Getting Started</b>	<b>7</b>
<b>3 Basics</b>	<b>9</b>
3.1 Manipulating the Scene . . . . .	9
3.1.1 World View . . . . .	10
3.1.2 Eye View . . . . .	11
3.2 3D Smoke . . . . .	11
3.3 Tracer Particles - Particle Files . . . . .	12
3.4 2D Shaded Contours - Slice Files . . . . .	14
3.5 2D Shaded Contours on Solid Surfaces - Boundary Files . . . . .	17
3.6 3D Contours - Isosurface Files . . . . .	20
3.7 Static Data - Plot3D Files . . . . .	22
3.8 Zone Fire Modeling Data - PLT Files . . . . .	22
<b>4 Advanced Features</b>	<b>25</b>
4.1 Manipulating the Scene Automatically - The Touring Option . . . . .	25
4.2 Clipping Scenes . . . . .	29
4.3 Setting Data Bounds . . . . .	31
4.4 3D Smoke Options . . . . .	32
4.5 Plot3D Viewing Options . . . . .	32
4.6 Display Options . . . . .	35
4.6.1 General . . . . .	35
4.6.2 Stereo . . . . .	35
4.7 Texture Maps . . . . .	36
4.8 Using Smokeview to Debug FDS Input Files . . . . .	38
4.9 Making Movies . . . . .	39
4.10 Annotating the Scene . . . . .	40
4.11 Creating Custom Devices . . . . .	41
4.11.1 Device File Format . . . . .	42

4.11.2	Geometric Objects . . . . .	46
4.11.3	Transformations . . . . .	47
4.12	Smokezip - Compressing 3D Smoke and Boundary Files . . . . .	48
<b>5</b>	<b>Summary</b>	<b>51</b>
	<b>References</b>	<b>54</b>
	<b>Appendices</b>	<b>54</b>
<b>A</b>	<b>Command Line Options</b>	<b>55</b>
<b>B</b>	<b>Menu Options</b>	<b>57</b>
B.1	Main Menu Items . . . . .	57
B.2	Load/Unload . . . . .	58
B.3	Show/Hide . . . . .	60
B.3.1	Geometry Options . . . . .	60
B.3.2	Animated Surface . . . . .	61
B.3.3	Particles . . . . .	61
B.3.4	Boundary . . . . .	61
B.3.5	Animated Vector Slice . . . . .	62
B.3.6	Animated Slice . . . . .	62
B.3.7	Plot3D . . . . .	62
B.3.8	Heat detectors, Sprinklers, Thermocouples . . . . .	63
B.3.9	Textures . . . . .	63
B.3.10	Labels . . . . .	63
B.4	Options . . . . .	63
B.4.1	Shades . . . . .	65
B.4.2	Units . . . . .	65
B.4.3	Rotation . . . . .	65
B.4.4	Max Frame Rate . . . . .	65
B.4.5	Render . . . . .	67
B.4.6	Font Size . . . . .	67
B.4.7	Zoom . . . . .	67
B.5	Dialogs . . . . .	67
B.6	Tours . . . . .	68
<b>C</b>	<b>Keyboard Shortcuts</b>	<b>69</b>
<b>D</b>	<b>File Formats</b>	<b>71</b>
D.1	Smokeview Preference File Format (.ini files) . . . . .	71
D.1.1	Color parameters . . . . .	72
D.1.2	Size parameters . . . . .	74
D.1.3	Time, Chop and value bound parameters . . . . .	74
D.1.4	Data loading parameters . . . . .	77
D.1.5	Viewing parameters . . . . .	79
D.1.6	Tour Parameters . . . . .	83
D.1.7	Realistic Smoke Parameters . . . . .	84
D.1.8	Zone Fire Modeling Parameters . . . . .	84

D.2	Smokeview Parameter Input File (.smv file) . . . . .	85
D.2.1	Geometry Keywords . . . . .	85
D.2.2	File Keywords . . . . .	87
D.2.3	Device (sensor) Keywords . . . . .	89
D.2.4	Miscellaneous Keywords . . . . .	90
D.3	Data File Formats (.s3d, .iso, .part, .sf, .bf, .q and CAD/.ge1 files) . . . . .	91
D.3.1	3D Smoke File Format . . . . .	91
D.3.2	Isosurface File Format . . . . .	92
D.3.3	Particle File Format . . . . .	92
D.3.4	Slice File Format . . . . .	93
D.3.5	Terrain File Format . . . . .	93
D.3.6	Boundary Files . . . . .	94
D.3.7	Plot3D Data . . . . .	95
D.3.8	CAD/GE1 file format . . . . .	95
<b>E</b>	<b>Frequently Asked Questions</b>	<b>97</b>
E.1	How can I keep up with new releases and other information about FDS and Smokeview? . . .	97
E.2	Can I run Smokeview on other machines? . . . . .	97
E.3	Smokeview doesn't look right on my computer. What's wrong? . . . . .	97
E.4	How do I make a movie of a Smokeview animation? . . . . .	98
E.5	Smokeview is running much slower than I expected. What can I do? . . . . .	98
E.6	I loaded a particle file but Smokeview is not displaying particles. What is wrong? . . . . .	98
E.7	Why do blockages not appear the way I defined them in my input file? ( invisible blockages are visible, smooth blockages are not smooth etc ) What is wrong? . . . . .	98



# List of Figures

1.1	FDS file overview . . . . .	2
3.1	Motion Dialog Box . . . . .	10
3.2	Smoke3D file snapshots of a townhouse kitchen fire. . . . .	13
3.3	Particle file snapshots of a townhouse kitchen fire. . . . .	14
3.4	Streak lines showing particle paths from 6 to 10 seconds in a townhouse kitchen fire. . . . .	15
3.5	Slice file snapshots of shaded temperature contours. . . . .	16
3.6	Slice file snapshots illustrating old and new method for coloring data. . . . .	17
3.7	Vector slice file snapshots of shaded vector plots. . . . .	18
3.8	Boundary file snapshots of shaded wall temperatures contours. . . . .	19
3.9	Boundary file snapshots showing ignited surfaces . . . . .	20
3.10	Isosurface file snapshots of temperature levels. . . . .	21
3.11	Plot3D contour and vector plot examples. . . . .	22
3.12	Plot3D isocontour example. . . . .	23
3.13	CFAST 6.0 Standard case showing upper layer and vent flow at 375 s. . . . .	23
4.1	Overhead view of the townhouse example showing the default <i>Circle</i> tour and a user defined tour. . . . .	26
4.2	Touring dialog boxes . . . . .	27
4.3	Tutorial examples for Tour option . . . . .	28
4.4	Clipping dialog box . . . . .	30
4.5	Two views of a multi-mesh case. . . . .	30
4.6	File/Bounds dialog box showing PLOT3D file options . . . . .	31
4.7	File/Bounds dialog box showing slice and boundary file options . . . . .	32
4.8	Ceiling Jet Visualization. . . . .	33
4.9	Dialog Box for setting 3D smoke options . . . . .	34
4.10	Dialog Box for setting miscellaneous Smokeview scene properties . . . . .	35
4.11	Stereo pair view of a townhouse kitchen fire. . . . .	35
4.12	Dialog box for activating the stereo view option. . . . .	36
4.13	Texture map example. . . . .	37
4.14	Blockage Edit Dialog Box . . . . .	39
4.15	Annotation example using the TICKS and LABEL keyword. . . . .	41
4.16	Device Object file format . . . . .	42
4.17	Instructions for drawing a sensor along with the corresponding Smokeview view. . . . .	43
4.18	Instructions for drawing an inactive and active heat detector along with the corresponding Smokeview view . . . . .	44
4.19	Smokeview view of devices defined in the global devices.svo file . . . . .	45
4.20	Compression and Autoload dialog box . . . . .	48

B.1	Main Menu . . . . .	58
B.2	Load/Unload Menu . . . . .	59
B.3	Geometry Menu . . . . .	60
B.4	Label Menu . . . . .	63
B.5	Option Menu . . . . .	64
B.6	Shades Menu . . . . .	64
B.7	Render Menu . . . . .	66
B.8	Dialogs Menu . . . . .	67
B.9	Tour Menu . . . . .	68



# List of Tables

3.1	Keyboard mappings for <i>eye centered</i> or first person scene movement . . . . .	12
3.2	Some output quantities used to generate slice and Plot3D visualizations. . . . .	18
3.3	Some output quantities used to generate boundary file visualizations. . . . .	20
D.1	Descriptions of parameters used by the Smokeview OBST keyword . . . . .	86
D.2	Descriptions of parameters used by the Smokeview VENT keyword . . . . .	88



# Chapter 1

## Introduction

### 1.1 Overview

Smokeview is a software tool designed to visualize numerical predictions generated by the NIST Fire Dynamics Simulator (FDS), a computational fluid dynamics (CFD) model of fire-driven fluid flow[2]. This report documents version 5 of Smokeview updating material found in Ref. [3]. For details on setting up and running FDS cases read the FDS User's guide[1].

FDS and Smokeview are used to model and visualize fire phenomena. However, FDS and Smokeview are not limited to fire simulation. For example, one may use FDS and Smokeview to model other applications such as contaminant flow in a building. Smokeview performs this visualization by displaying time dependent tracer particle flow, animated contour slices of computed gas variables and surface data. Smokeview also presents contours and vector plots of static data anywhere within a simulation scene at a fixed time. Several examples using these techniques to investigate fire incidents are documented in Refs. [4, 5, 6, 7].

Normally Smokeview is used in a post-processing step to visualize FDS data after a calculation has been completed. Smokeview may also be used during a calculation to monitor a simulation's progress and before a calculation to setup FDS input files more quickly, one can then use Smokeview to edit or create blockages by specifying the size, location and/or material properties.

Figure 1.1 gives an overview of how data files used by FDS, Smokeview and Smokezip are related. A typical procedure for using FDS and Smokeview is to:

1. Set up an FDS input file.
2. Run FDS. FDS then creates one or more output files used by Smokeview to visualize the case.
3. Run Smokeview to analyze the output files generated by step 2. by either double-clicking the file named `casename.smv` with the mouse (on the PC) or by typing `smokeview casename` at a command line. Smokeview may also be used to create new blockages and modify existing ones. The blockage changes are saved in a new FDS input data file.

This publication documents step 3. Steps 1 and 2 are documented in the FDS User's Guide [1].

Menus in Smokeview are activated by clicking the right mouse button anywhere in the Smokeview window. Data files may be visualized by selecting the desired `Load/Unload` menu option. Other menu options are discussed in Appendix B. Many menu commands have equivalent keyboard shortcuts. These shortcuts are listed in Smokeview's `Help` menu and are described in Appendix C. Visualization features not controllable through the menus may be customized by using the Smokeview preference file, `smokeview.ini`, discussed in Appendix D.1.

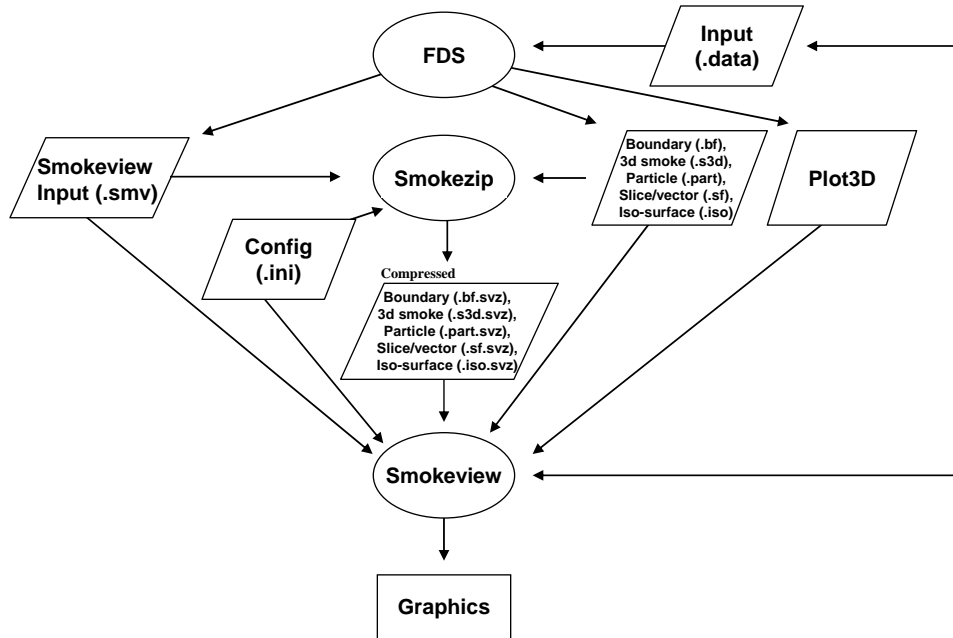


Figure 1.1: Diagram illustrating files used and created by the NIST Fire Dynamics Simulator (FDS), Smokezip and Smokeview.

Smokeview is written in C and Fortran 90 and consists of about 70,000 lines of code. The C portion of Smokeview visualizes the data while the Fortran 90 portion reads in the data generated by FDS (also written in Fortran 90). Smokeview uses the 3D graphics library OpenGL[8] and the Graphics Library Utility Toolkit (GLUT)[9]. Smokeview uses the GLUT software library so that most of the development effort can be spent implementing the visualizations rather than creating an elaborate user interface. Smokeview uses a number of auxiliary libraries to implement image capture (GD[10, 11], PNG[12], JPEG[13]), image and general file compression (ZLIB[14]) and dialog creation (GLUI[15]). Each of these libraries is portable running under UNIX, LINUX and Windows 9x/2000/XP allowing Smokeview to run on these platforms as well.

## 1.2 Features

Smokeview is a program designed to visualize numerical calculations generated by the Fire Dynamics Simulator. Smokeview visualizes both dynamic and static data. Dynamic data is visualized by animating particle flow (showing location and *values* of tracer particles), 2D contour slices (both within the domain and on solid surfaces) and 3D level surfaces. 2D contour slices can also be drawn with colored vectors that use velocity data to show flow direction, speed and value. Static data is visualized similarly by drawing 2D contours, vector plots and 3D level surfaces. Smokeview features in more detail include:

**3D Smoke** Smoke, fire and sprinkler spray are displayed realistically using a series of partially transparent planes. The smoke transparencies are determined by using smoke densities computed by FDS. The fire and sprinkler spray transparencies are determined by using a heuristic based on heat release rate and water density data, again computed by FDS. Various settings for the 3d smoke option may be set using the new *3D Smoke* dialog box found in the Dialogs menu.

**Animated Isosurfaces** Isosurface or 3D level surface animations may be used to represent flame boundaries, layer interfaces or various other gas phase variables. Multiple isocontours may be stored in one

file, allowing one to simultaneously view several isosurface levels for the same variable.

**Color Contours** Animated 2D shaded color contour plots are used to visualize gas phase information, such as temperature or density. The contour plots are drawn in horizontal or vertical planes along any coordinate direction. Contours can also be drawn in shades of gray.

Animated 2D shaded color contour plots are also used to visualize solid phase quantities such as radiative flux or heat release rate per unit area.

**Animated Flow Vectors** Flow vector animations, though similar to color contour animations (the vector colors are the same as the corresponding contour colors), are better than solid contour animations at highlighting flow features.

**Particle Animations** Lagrangian or moving particles can be used to visualize the flow field. Often these particles represent smoke or water droplets.

**Data Mining** The user can analyze and examine the simulated data by altering its appearance to more easily identify features and behaviors found in the simulation data. One may flip or reverse the order of colors in the colorbar and also click in the colorbar and slide the mouse to highlight data values in the scene. These options may be found under `Options/Shades`.

The user may click in the time bar and slide the mouse to change the simulation time displayed. One use for the time bar and color bar selection modes might be to determine when smoke of a particular temperature enters a room.

**Multimesh Geometry** Smokeview shows the multi-mesh geometry new in FDS 3, and gives the user control over viewing certain features in a given mesh. Using the `LOAD/UNLOAD > MULTI-SLICE` menu item one may now load multiple slices simultaneously. These slices are ones lying in the same plane (within a grid cell) across multiple meshes.

**Scene Clipping** It is often difficult to visualize slice or boundary data in complicated geometries due to the number of obstructed surfaces. Interior portions of the scene may now be seen more easily by *clipping* part of the scene away.

**Scene Motion** The motion dialog box has been enhanced to allow more precise control of scene movement and orientation. Cursor keys have been mapped to scene translation/rotation to allow easy navigation within the scene.

**Texture Mapping** Jpeg or PNG image files may be applied to a blockage, vent or enclosure boundary. This is called texture mapping. This allows one to view Smokeview scenes more realistically. These image files may be obtained from the internet, a digital camera, a scanner or from any other source that generates these file formats. Image files used for texture mapping should be *seamless*. A seamless texture as the name suggests is periodic in both horizontal and vertical directions. This is an especially important feature when textures are tiled or repeated across a blockage surface.

**Blockage Editing** The Blockage Editing dialog box has been enhanced to allow one to assign material properties to selected blockages. Comment labels found in the FDS input file may be viewed or edited. Any text appearing after the closing “/” in an `&OBST` line is treated as a comment by Smokeview.

**Annotating Cases** The two keywords, `LABEL` and `TICKS` are used to help document Smokeview output. The `LABEL` keyword allows one to place colored labels at specified locations at specified times. The `TICK` keyword places equally spaced tick marks between specified bounds. These marks along with `LABEL` text may be used to document length scales in the scene.

**Scene Movement** The first person or eye view mode for moving has been enhanced to allow one to move through a scene more realistically by preventing one from moving through blockages and allowing one to climb stairs, similar to the types of motion permitted in modern computer games. Using the cursor keys and the mouse, one can move through a scene while the fire is progressing performing tasks related to a training exercise.

**Transparent Blockages** One may now specify blockages with partially transparent colors using the `COLOR` or `RGB` keyword on either the `&OBST` or `&SURF` lines to specify the color and `TRANSPARENCY` on these same lines to specify the transparency amount (0.0 for completely transparent, 1.0 for completely opaque). This allows one to see through a solid enclosure by defining *windows* to be partially transparent.

**Virtual Tour** A series of checkpoints or key frames specifying position and view direction may be specified. A smooth path is computed using Kochanek-Bartels splines[16] to go through these key frames so that one may control the position and view direction of an observer as they move through the simulation. One can then see the simulation as the observer would. This option is available under the `Tour` menu item. Existing tours may be edited and new tours may be created using the *Tour* dialog box found in the `Dialogs` menu. Tour settings are stored in the local configuration file (`casename.ini`).

**Data Chopping** - The `File/Bounds Settings...` dialog box has been enhanced to allow one to chop or hide data in addition to setting bounds. One use of this feature would be to more easily visualize a ceiling jet by hiding ambient temperature data (data below a prescribed temperature).

**Time Averaging** - The `File/Bounds Settings...` dialog box has also been enhanced to allow one to time average slice file data. Data may be smoothed over a user selectable time interval.

**Data Compression** - An option has been added to the `LOAD/UNLOAD` menu to compress 3d smoke and boundary files. The option shells out to the program `smokezip` which runs in the background enabling one to continue to use Smokeview while files are compressing.

## 1.3 What's New

Several features have been added to Smokeview to improve the user's ability to visualize fire scenarios.

**3D slice files** The user may now visualize a 3D region of data using slice files. Slices may be moved from one plane to the next just as with `PLOT3D` files (using up/down cursor keys or page up/page down keys). Data for 3D slice files are generated by specifying a 3D rather than a 2D region with the `&SLCF` keyword.

**color display** The display of color contours used to draw slice, boundary and `PLOT3D` files has been improved. The colors are now crisper and sharper, more accurately representing the underlying data. This is most noticeable when selecting the colorbar with the mouse. As before this causes a portion of the colorbar to turn black and the corresponding region in the scene to also turn black. Now the black color is accurate to the pixel so this feature could be used to highlight regions of interest.

**streak lines** The new particle file format used in FDS 5 allows Smokeview to display particles as streak lines (a particle drawn where it has been for a short period of time in the past). Streak lines are a good method for displaying motion with *still* pictures.

**general objects** A new method for drawing objects (an object being a heat detector, smoke detector, sprinkler sensor *etc.*) has been implemented in Smokeview 5. These objects look more realistic. Objects are specified in a data file rather than in Smokeview as C code. This allows the user to customize the *look* and *feel* of the objects (say to match the types of detectors/sprinklers that are being used) without requiring code changes in Smokeview.

**stereo views** A method for displaying stereo/3D images has been implemented that does not require any specialized equipment such as shuttered glasses or quad buffered enabled video cards. Stereo pair images are displayed side by side after invoking the option with the *Stereo* dialog box or pressing the "S" key (upper case). A 3D view may be seen by relaxing the eyes allowing them to merge the two images into one.





## Chapter 2

# Getting Started

Smokeview may be obtained at <http://fire.nist.gov/fds>. This site gives links to a Setup program for PC installation. It also contains documentation for Smokeview and FDS, sample FDS calculations, software updates and links for requesting feedback about the software.

After obtaining the setup program, install Smokeview on the PC by either entering the setup program name from the Windows `Start/Run...` menu or by double-clicking the downloaded Smokeview setup program. The setup program then steps through the program installation. It copies the FDS and Smokeview executables, sample cases, documentation and the Smokeview preference file `smokeview.ini` to the default directory `C:\nist\fds`. The setup program also defines PATH variables and associates the `.smv` file extension to the Smokeview program so that one may either type Smokeview at any command line prompt or double click on any `.smv` file. Smokeview uses the OpenGL graphics library which is a part of all Windows distributions.

Most computers purchased today are perfectly adequate for running Smokeview. For Smokeview it is more important to obtain a fast graphics card than a fast CPU. If the computer will run both FDS and Smokeview then it is important to obtain a fast CPU as well. For example, the townhouse case found at <http://fire.nist.gov/fds/svsamples/thouse3d.data> consists of about 180,000 grid cells and is used in many of the Figures in this report, required 5.0 hours of CPU time on a 3.6 GHZ Xeon Windows XP system. Cases with more grid cells and longer simulation times (the townhouse case simulated 300 S of smoke flow) would clearly benefit from a faster CPU and more memory which are now relatively inexpensive.



## Chapter 3

# Basics

Smokeyview may be started on the PC by double-clicking the file named `casename.smv` where `casename` is the name specified by the `CHID` keyword defined in the FDS input data file. Menus are accessed by clicking with the right mouse button. The `Load/Unload` menu may be used to read in the data files to be visualized. The `Show/Hide` menu may be used to change how the visualizations are presented. For the most part, the menu choices are self explanatory. Menu items exist for showing and hiding various simulation elements, creating screen dumps, obtaining help *etc.* Menu items are described in Appendix B.

To use Smokeyview from a *command line*, open a command shell on a PC or a UNIX shell on a UNIX workstation. Then change to the directory containing the FDS case to be viewed and type:

```
smokeview casename
```

where `casename` is the name specified by the `CHID` keyword defined in the FDS input data file. Data files may be loaded and options may be selected by clicking the right mouse button and picking the appropriate menu item.

Smokeyview opens two windows, one displays the scene and the other displays status information. Closing either window will end the Smokeyview session. Multiple copies of Smokeyview may be run simultaneously if the computer has adequate resources.

Normally Smokeyview is run during an FDS run, after the run has completed and as an aid in setting up FDS cases by visualizing geometric components such as blockages, vents, sensors, *etc.* One can then verify that these modelling elements have been defined and located as intended. One may select the color of these elements using color parameters in the `smokeview.ini` to help distinguish one element from another. Smokeyview.ini file entries are described in section D.1.

Although specific video card brands cannot be recommended, they should be *high-end* due to Smokeyview's intensive graphics requirements. These requirements will only increase in the future as more features are added. A video card designed to perform well for *fancy* computer games should do well for Smokeyview. Some apparent bugs in Smokeyview have been found to be the result of problems found in video cards on older computers.

### 3.1 Manipulating the Scene

The scene may be manipulated from two points of view, a world or global view and a first person or *eye* view. These views may be switched by pressing the “e” key or by selecting the appropriate radio button in the Motion/View dialog box.

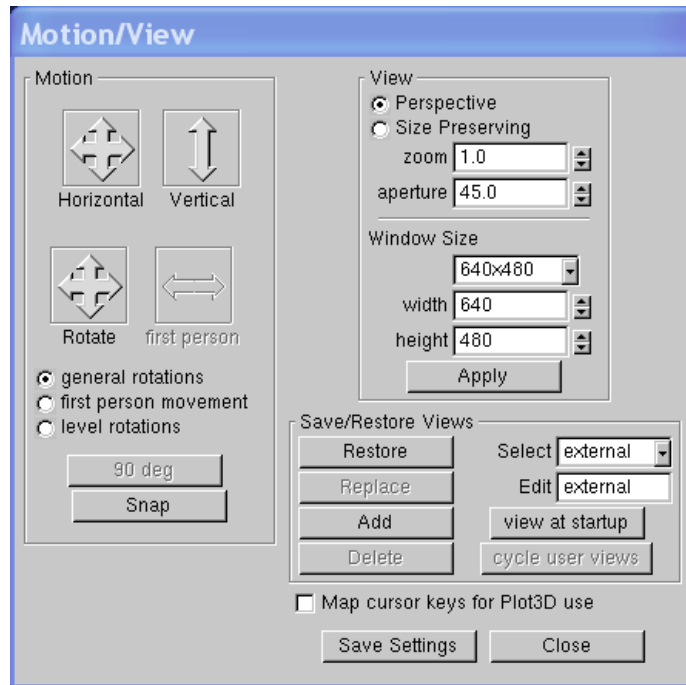


Figure 3.1: Motion Dialog Box. Rotate or translate the scene by clicking an *arrow* and dragging the mouse. The Motion Dialog Box is invoked by selecting `Dialogs>Motion/View`

### 3.1.1 World View

The scene may be rotated or translated while in world view, either directly with the mouse or by using controls contained in the scene movement dialog box. This dialog box is opened from the `Dialogs>Motion/View` menu item and is illustrated in Figure 3.1. Clicking on the scene and dragging the mouse horizontally, vertically or a combination of both results in scene rotation or translation depending upon whether `CTRL` or `ALT` is depressed or not during mouse movement.

In particular, when modifier keys are not depressed, horizontal or vertical mouse movement results in scene rotation parallel to the XY or YZ plane, respectively. Pressing the CTRL and ALT modifier keys while moving the mouse results in scene movement in the following ways:

**CTRL key depressed** Horizontal mouse movement results in scene translation from side to side along the X axis. Vertical mouse movement results in scene translation into and out of the computer screen along the Y axis.

**ALT key depressed** Vertical mouse movement results in scene translation along the Z axis. Horizontal mouse movement has no effect on the scene while the `ALT` key is depressed.

The Motion/View dialog box, illustrated in Figure ?? may be used to move the scene in a more controlled manner. For example, buttons in the *Motion* region allows one to translate or rotate the scene. The `Horizontal` button allows one to translate the scene horizontally in a left/right or in/out direction while the `Vertical` button allows one to translate the scene in an up/down direction.

Controls in the *View* region of the Motion/View dialog box allow one to change the scene magnification or zoom factor and the projection method used to draw objects (perspective or isometric) These two

projection methods differ in how objects are displayed at a distance. A perspective projection for-shortens or draws an object smaller when drawn at a distance. An isometric projection on the other hand draws an objects the same size regardless of where it is drawn in the scene. The projection method desired may be selected using the *View* section of the Motion/View dialog box.

The  and  edit boxes allow one to change the magnification of the scene or equivalently the angle of view across the scene. The relation between these two parameters is given by

$$\text{zoom} = \tan(45^\circ/2)/\tan(\text{aperture}/2)$$

A default aperture of  $45^\circ$  is chosen so that Smokeview scenes have a normal perspective.

may be used to reset the scene back to either an external, internal (to the scene), or previously saved viewpoint.

**Select Rotation Center** A pull down list appears in multi-mesh cases allowing one to change the rotation center. Therefore one could rotate the scene about the center of the entire physical domain or about the center of any one particular mesh. This is handy when meshes are defined far apart.

**Rotation Buttons** Rotation buttons are enabled or disabled as appropriate for the mode of scene motion. For example, if *about world center - level rotations* has been selected, then the  and  buttons are disabled. A rotation button labelled  has been added to allow one to rotate 90 degrees while in *eye center* mode. This is handy when one wishes to move down a long corridor precisely parallel to one of the walls. The first click of *90 deg* snaps the view to the closest forward or side direction while each additional click rotates the view 90 degrees clockwise.

**Viewpoint Buttons** A viewpoint is the combination of a location and a view direction. Several new buttons have been added to this dialog box to save and restore viewpoints. The scene is manipulated to the desired orientation then stored the **Add** button to add the new viewpoint or the **Replace** button to replace this viewpoint with the currently selected one. To change the view to a currently stored viewpoint, use the **Select** listbox to select the viewpoint and the **Restore** button to activate it. The **Delete** button may be used to remove a viewpoint from the stored list. The **Edit** text box may be used to change the name or label for the selected viewpoint. The **view at startup** button is used to specify the viewpoint that should be set when Smokeview first starts up.

### 3.1.2 Eye View

Radio buttons in the Motion/View dialog box allow one to toggle between world, eye centered and world level rotation scene movement modes. These modes may also be changed by using the “e” key. When in *eye center* mode, several key mappings have been added, inspired by popular computer games, to allow for easier movement within the scene. For example, the up and down cursor keys allow one to move forward or backwards. The left and right cursor keys allow one to rotate left or right. Other keyboard mappings are described in Table 3.1.

## 3.2 3D Smoke

FDS generates several data files visualized by Smokeview. Each file type may be loaded or unloaded using the  menu described in Appendix B.2. Visualizations produced by these data files

Table 3.1: Keyboard mappings for *eye centered* or first person scene movement

Key	Description
up/down cursor w/s	move forward/backward
ALT + left/right cursor a/d	slide left/right
ALT + up/down cursor	move up/down
left/right cursor	rotate left/right
Page Up/Down	look up/down
Home	look level
j	toggle between crawling and walking
Pressing the SHIFT key while moving, sliding or rotating results in a 4x speedup of these actions.	

are described in this and the following sections. The format used to store each of the data files is given in Appendix D.3. The FDS input data file used to generate the following examples may be found at <http://fire.nist.gov/fds/svsamples/thouse3d.data>.

Visualizing smoke realistically is a daunting challenge for at least three reasons. First, the storage requirements for describing smoke can easily exceed the disk capacities of present 32 bit operating systems such as Linux, *i.e.* file sizes can easily exceed 2 gigabytes. Second, the computation required both by the CPU and the video card to display each frame can easily exceed 0.1 s, the time corresponding to a 10 frame/s display rate. Third, the physics required to describe smoke and its interactions with itself and surrounding light sources is complex and computationally intensive. Therefore, approximations and simplifications are required to display smoke rapidly.

Smoke visualization techniques such as tracer particles or shaded 2D contours are useful for quantitative analysis but not suitable for virtual reality applications, where displays need to be realistic and fast as well as accurate. The approach taken by Smokeview is to display a series of parallel planes. Each plane is colored black (for smoke) with transparency values pre-computed by FDS using time dependent soot densities computed by FDS corresponding to the grid spacings of the simulation. The transparencies are adjusted in real time by Smokeview to account for differing path lengths through the smoke as the view direction changes. The graphics hardware then combines the planes together to form one image. For some view angles the smoke *planes* may be visible. Changing the viewing angle a small amount will normally make the smoke appear more uniform.

Fire by default is colored a dark shade of orange wherever the computed heat release rate per unit volume exceeds a user-defined cutoff value. The visual characteristics of fire are not automatically accounted for. The user though may use the **3D Smoke** dialog box to change both the color and transparency of the fire for fires that have *non-standard* colors and opacities.

Figure 3.2 illustrates a visualization of realistic smoke.

### 3.3 Tracer Particles - Particle Files

Particle files contain the locations of tracer particles used to visualize the flow field. Figure 3.3 shows several snapshots of a developing kitchen fire visualized by using particles where particles are colored black. If present, sprinkler water droplets would be colored blue. Particles are stored in files ending with

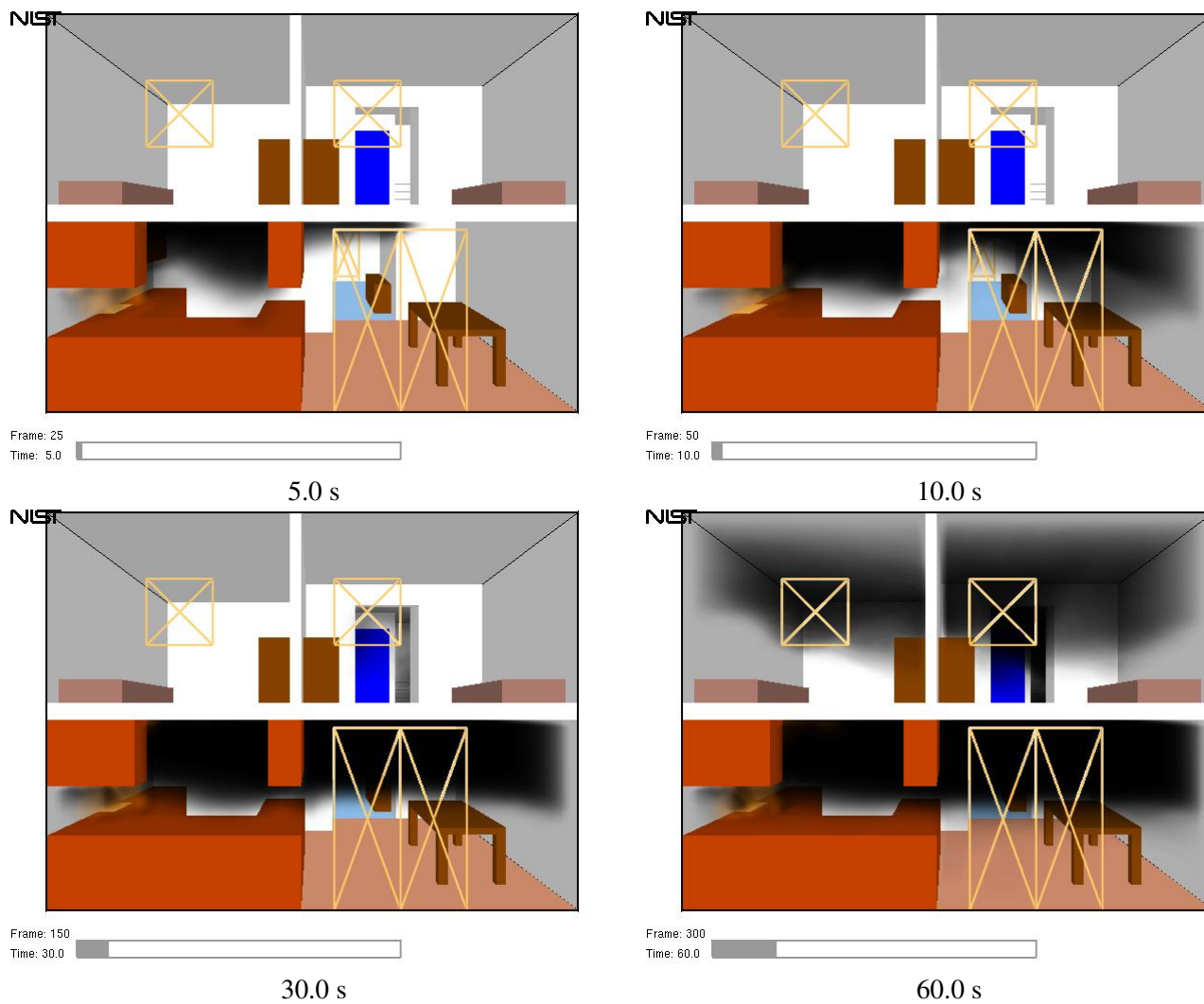


Figure 3.2: Cross section of a townhouse showing first and second floor. Smoke3d file snapshots at various times in a simulation of a townhouse kitchen fire.

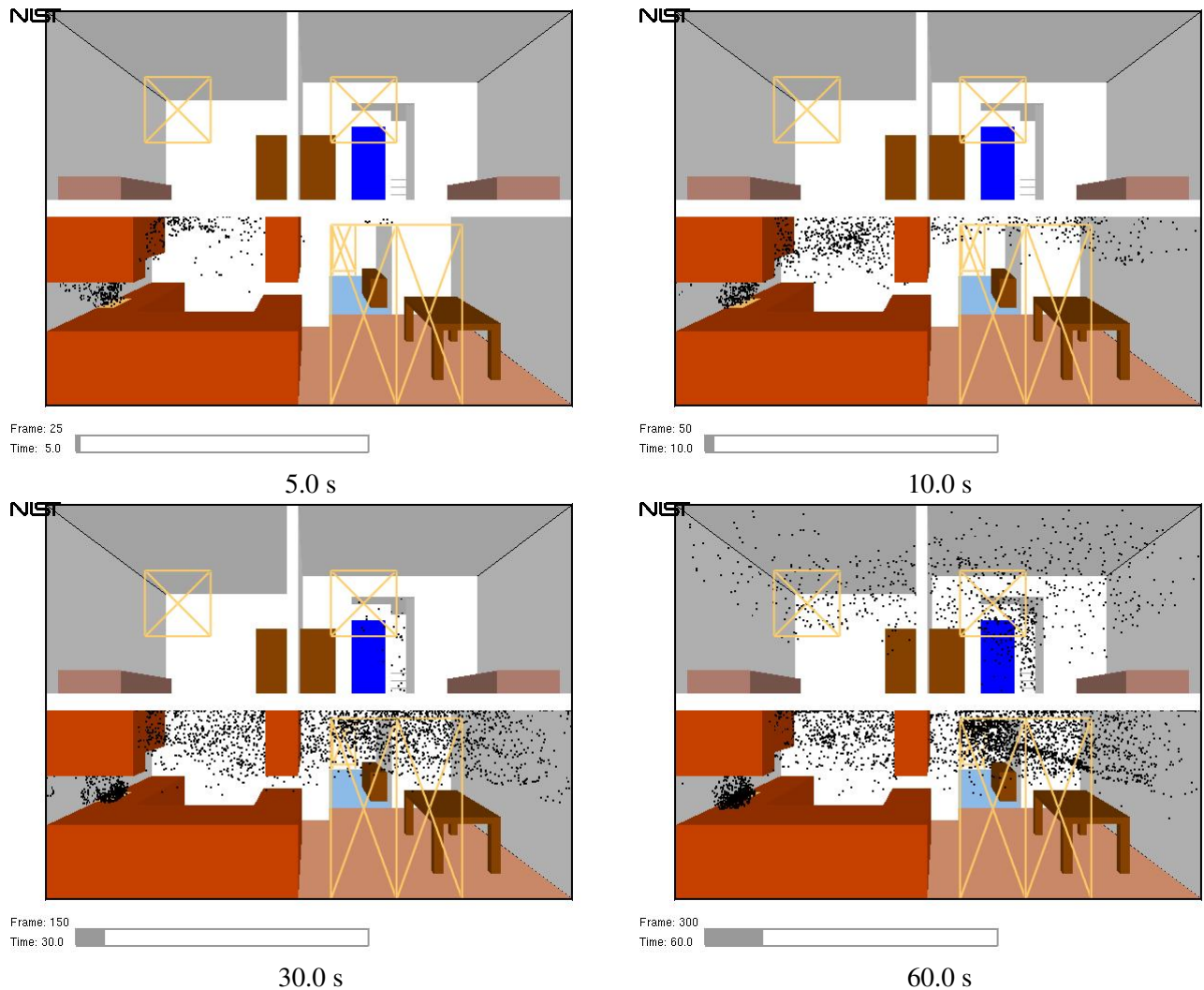


Figure 3.3: Cross section of a townhouse showing first and second floor. Particle file snapshots at four times during the simulation of a townhouse kitchen fire.

the extension `.part` and are displayed by selecting the entry from the `Load/Unload` menu. Figure 3.4 shows a snapshot of the same kitchen fire using streak lines instead of particles. The streaks begin at 6 s and end at 10 s. This is a good way to show motion using a static image.

### 3.4 2D Shaded Contours - Slice Files

Slice files contain results recorded within a rectangular array of grid points at each recorded time step. Continuously shaded contours are drawn for simulation quantities such as temperature, gas velocity and heat release rate. All or part of a plane is selected when setting up the FDS input data file. Figure 3.5 shows several snapshots of a vertical animated slice where the slice is colored according to gas temperature. Slice files have file names with extension `.sf` and are displayed by selecting the desired entry from the `Load/Unload` menu.

To specify in FDS a vertical slice 1.5 m from the  $y = 0$  boundary colored by temperature, use the line:



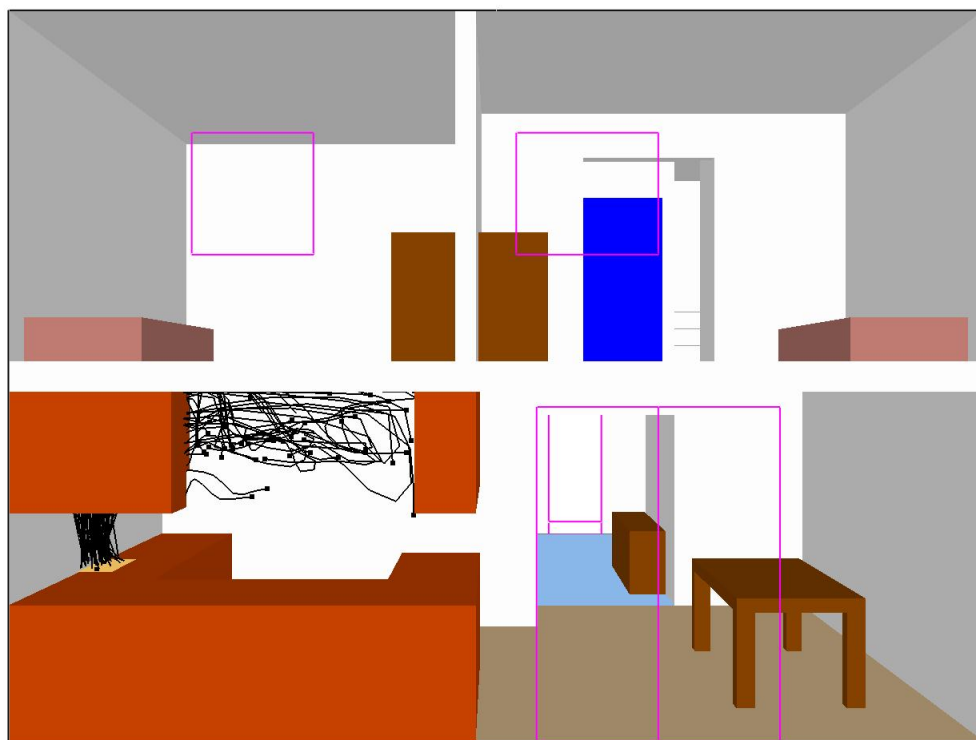


Figure 3.4: Cross section of a townhouse showing first and second floor. Streak lines showing particle paths from 6 to 10 seconds in a townhouse kitchen fire.

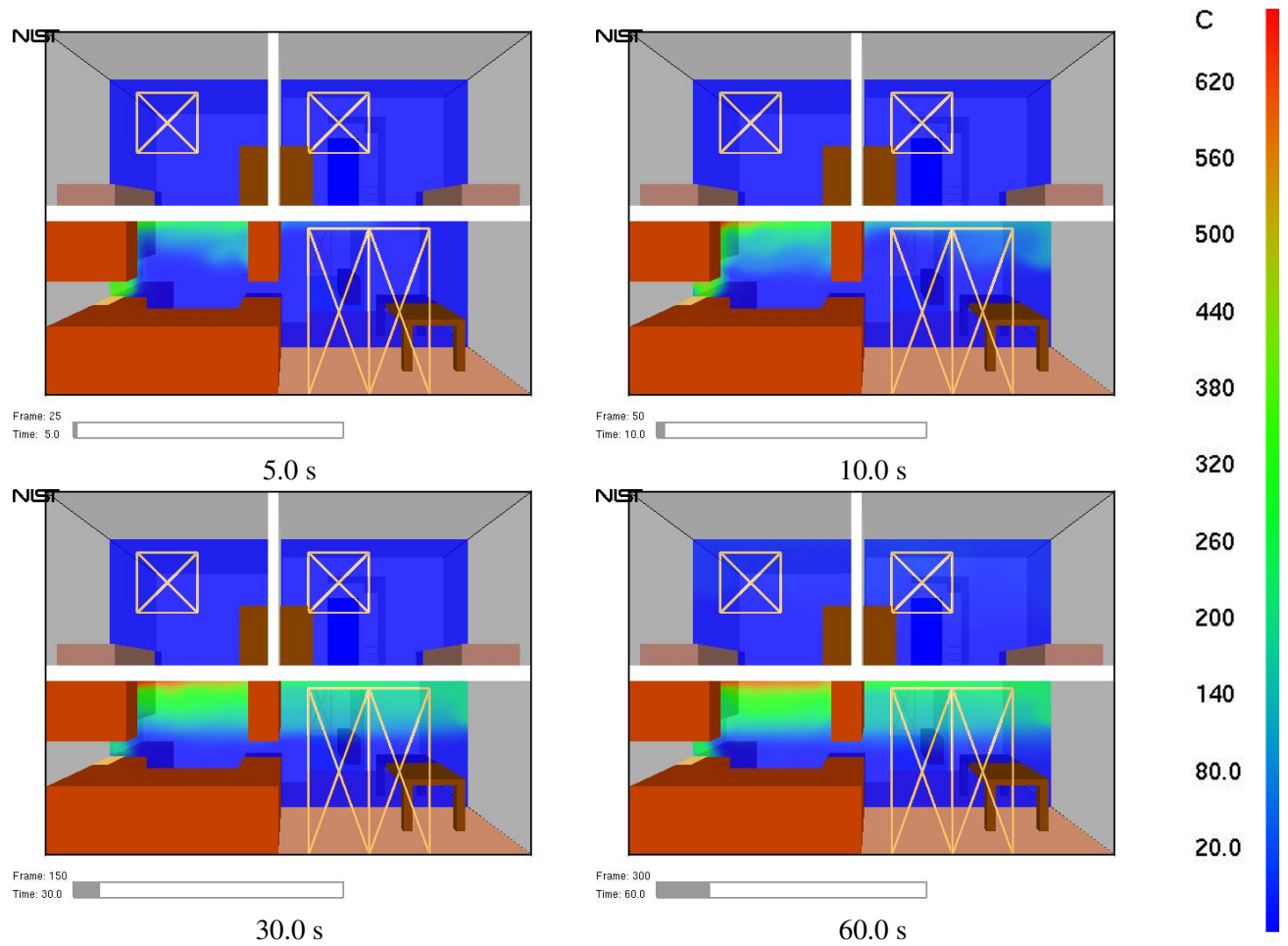


Figure 3.5: Slice file snapshots of shaded temperature contours at various times in a simulation. These contours were generated by adding “&SLCF PBX=1.5, QUANTITY='TEMPERATURE' /” to the FDS input file.

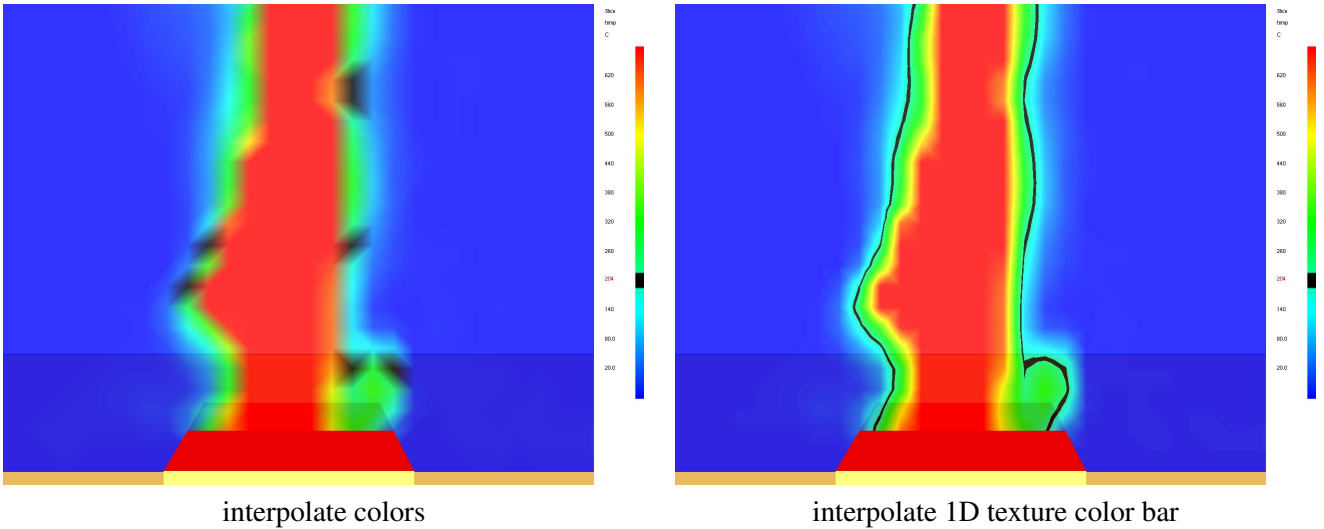


Figure 3.6: Slice file snapshots illustrating old and new method for coloring data.

```
&SLCF PBY=1.5 QUANTITY='TEMPERATURE' /
```

Other quantities that can be specified are given in Table 3.2. A more complete list may be found in Ref. [17].

**New data coloring** Smokeview uses a new more precise method for coloring data occurring in slice, boundary and PLOT3D files. This is illustrated in Figure 3.6. The colors are now crisper and sharper, more accurately representing the underlying data. This is most noticeable when selecting the colorbar with the mouse. As before this causes a portion of the colorbar to turn black and the corresponding region in the scene to also turn black. Now the black color is accurate to the pixel so this feature could be used to highlight regions of interest. Data is colored using 1D textures rather than coloring vertices.

**3D Slice Files** The user may now visualize a 3D region of data using slice files. The slices may be moved from one plane to the next just as with PLOT3D files (using up/down cursor keys or page up/page down keys). To specify in FDS a cube of data from 1.0 to 2.0 in each of the X, Y and Z directions, use the line:

```
&SLCF XB=1.0,2.0,1.0,2.0,1.0,2.0 QUANTITY='TEMPERATURE' /
```

Animated vectors are displayed using data contained in two or more slice files. The direction and length of the vectors are determined from the *U*, *V* and/or *W* velocity slice files. The vector colors are determined from the file (such as temperature) selected from the Load/Unload menu. Figure 3.7 shows a sequence of vector slices corresponding to the shaded temperature contours found in Figure 3.5.

To generate the extra velocity files needed to view vector animations, add `VECTOR=.TRUE.` to the above `&SLCF` line to obtain:

```
&SLCF PBY=1.50,QUANTITY='TEMPERATURE',VECTOR=.TRUE. /
```

### 3.5 2D Shaded Contours on Solid Surfaces - Boundary Files

Boundary files contain simulation data recorded at blockage or wall surfaces. Continuously shaded contours are drawn for quantities such as wall surface temperature, radiative flux, *etc.* Figure 3.8 shows sev-

Table 3.2: Some output quantities used to generate slice and Plot3D visualizations. A more complete list may be found in Ref.[17]

Quantity	Description	Units
DENSITY	density	$\text{kg/m}^3$
TEMPERATURE	gas temperature	$^{\circ}\text{C}$
U-VELOCITY	velocity component	m/s
V-VELOCITY	velocity component	m/s
W-VELOCITY	velocity component	m/s
VELOCITY	velocity magnitude	m/s

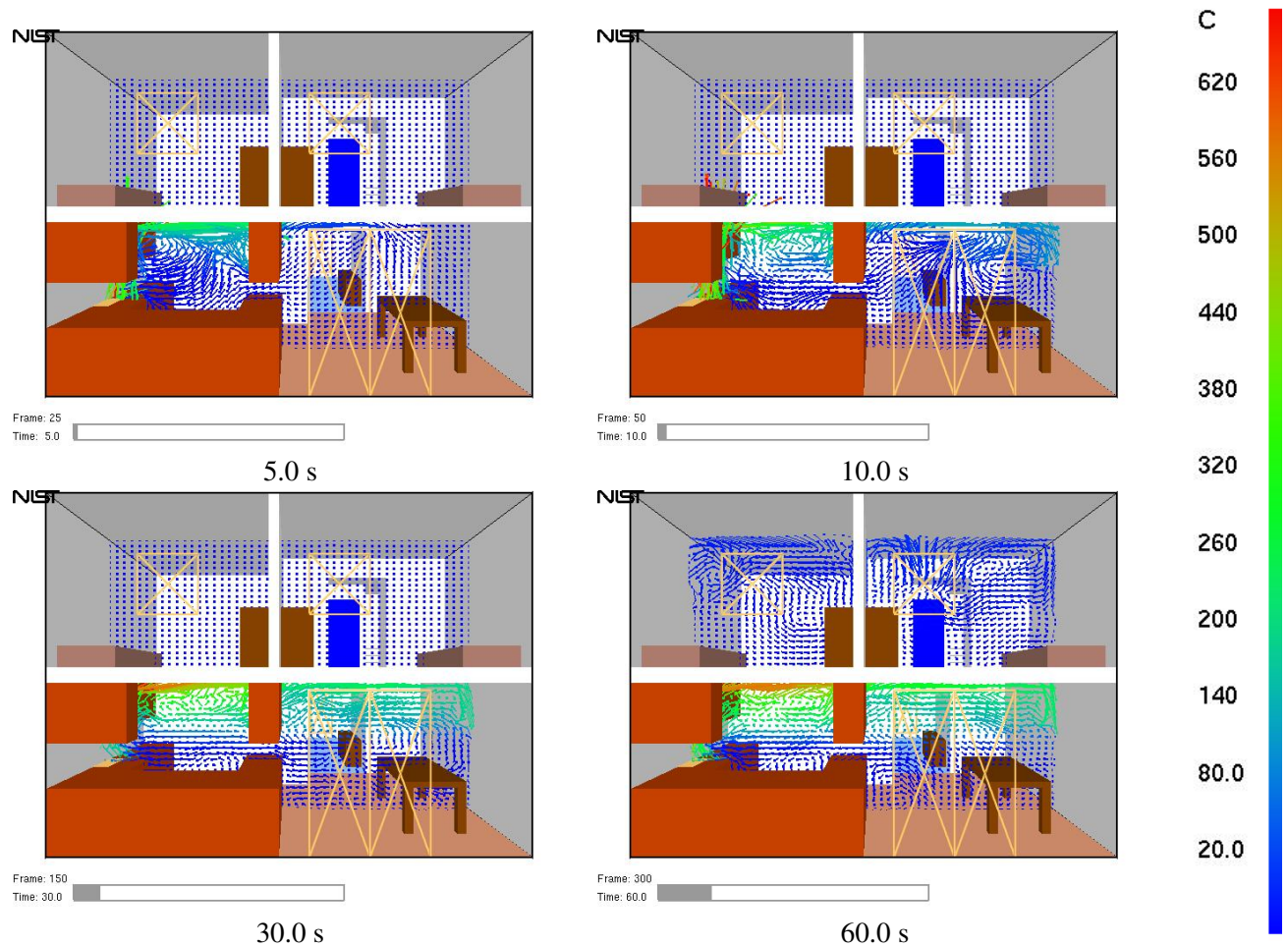


Figure 3.7: Vector slice file snapshots of shaded vector plots. These vector plots were generated by using “&SLCF PBX=1.5, QUANTITY='TEMPERATURE', VECTOR=.TRUE. /”.

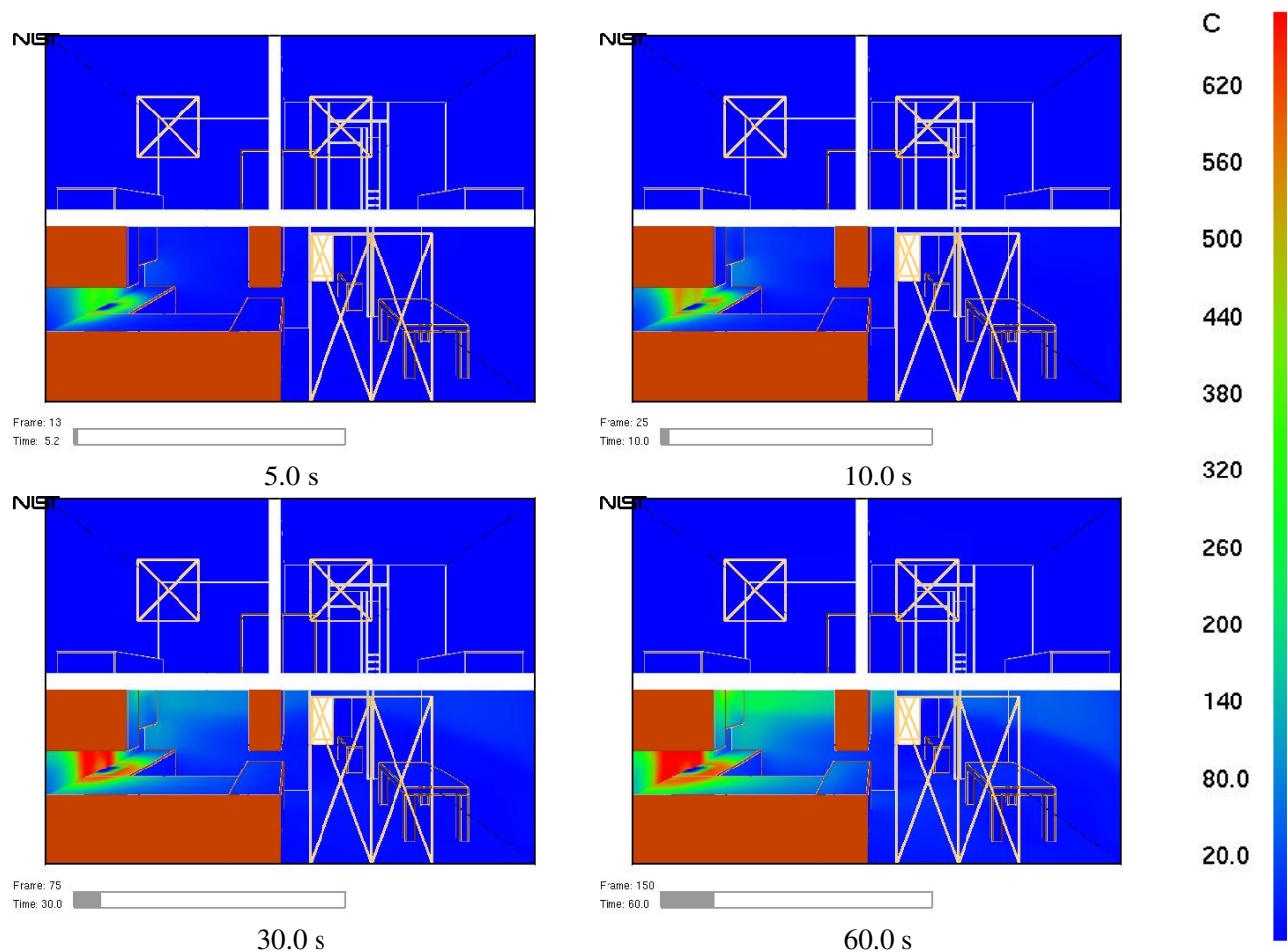


Figure 3.8: Boundary file snapshots of shaded wall temperatures. These snapshots were generated by using “&BNDF QUANTITY='WALL\_TEMPERATURE' /”.

eral snapshots of a boundary file animation where the surfaces are colored according to their temperature. Boundary files have file names with extension .bf and are displayed by selecting the desired entry from the  menu.

A boundary file containing wall temperature data may be generated by using:

```
&BNDF 'WALL_TEMPERATURE' /
```

Loading a boundary file is a memory intensive operation. The entire boundary file is read in to determine the minimum and maximum data values. These bounds are then used to convert four byte floats to one byte color indices. To drastically reduce the memory requirements, simply specify the minimum and maximum data bounds using the  dialog box. This should be done before loading the boundary file data. When this is done, memory for the boundary file data is allocated for only one time step rather than for all time steps.

**Highlighting Ignited Surfaces** Wall temperature boundary file data may be colored black where temperatures exceed the ignition temperature of the underlying surface where the ignition temperature is defined on the FDS &SURF line. This is activated by selecting the *Show Ignition* check box in the Set Bounds dialog

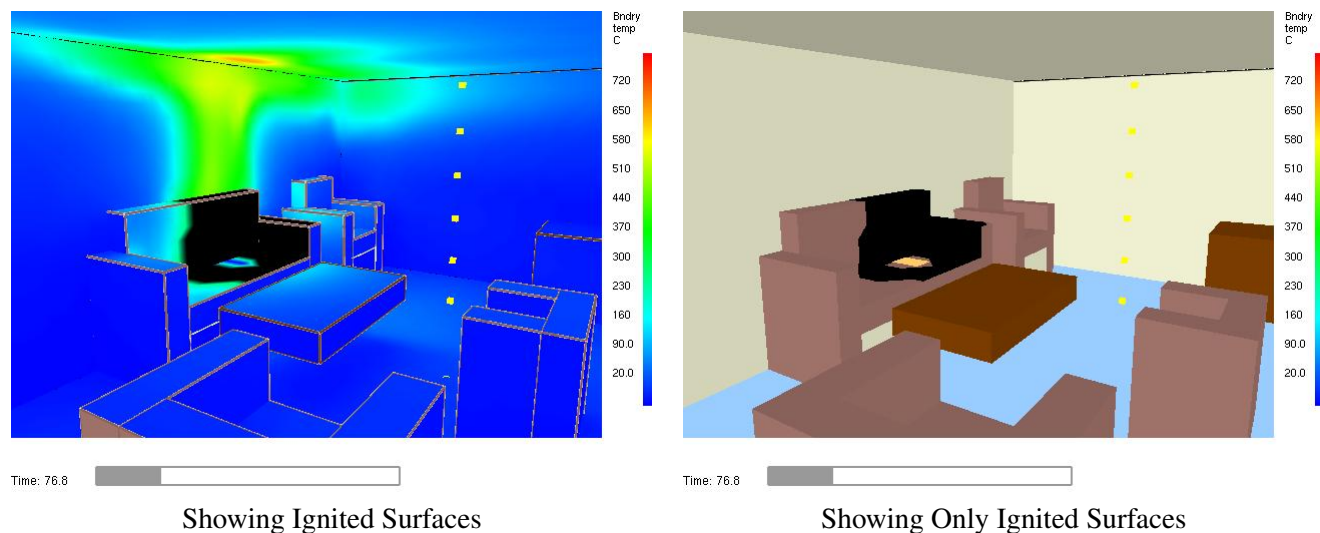


Figure 3.9: Boundary file snapshots showing ignited surfaces

Table 3.3: Some output quantities used to generate boundary file visualizations. A more complete list may be found in Ref.[17]

Quantity	Description	Units
RADIATIVE_FLUX	radiative flux	kW/m <sup>2</sup>
CONVECTIVE_FLUX	convective flux	kW/m <sup>2</sup>
WALL_TEMPERATURE	wall temperature	°C

box. If one also selects the *how Only Ignition* check box then the ignited material is drawn black but the rest of the boundary file is not drawn.

Table 3.3 gives other quantities that can be specified on a &BNDF line. A more complete list of quantities is given in Ref. [17].

### 3.6 3D Contours - Isosurface Files

The surface where a quantity such as temperature attains a given value is called an isosurface. An isosurface is also called a level surface or 3D contour. Isosurface files contain data specifying isosurface locations for a given quantity at one or more levels. These surfaces are represented as triangles. Isosurface files have file names with extension .iso and are displayed by selecting the desired entry from the Load/Unload menu.

Isosurfaces are specified in the FDS input file with the &ISOF keyword. To specify isosurfaces for temperatures of 30°C and 100°C as illustrated in Figure 3.10 add the line:

```
&ISOF QUANTITY='TEMPERATURE',VALUE(1)=30.0,VALUE(2)=100.0 /
```

to the FDS input file. Other quantities may be specified using values given in Table 3.2. A more complete list may be found in Ref. [17]

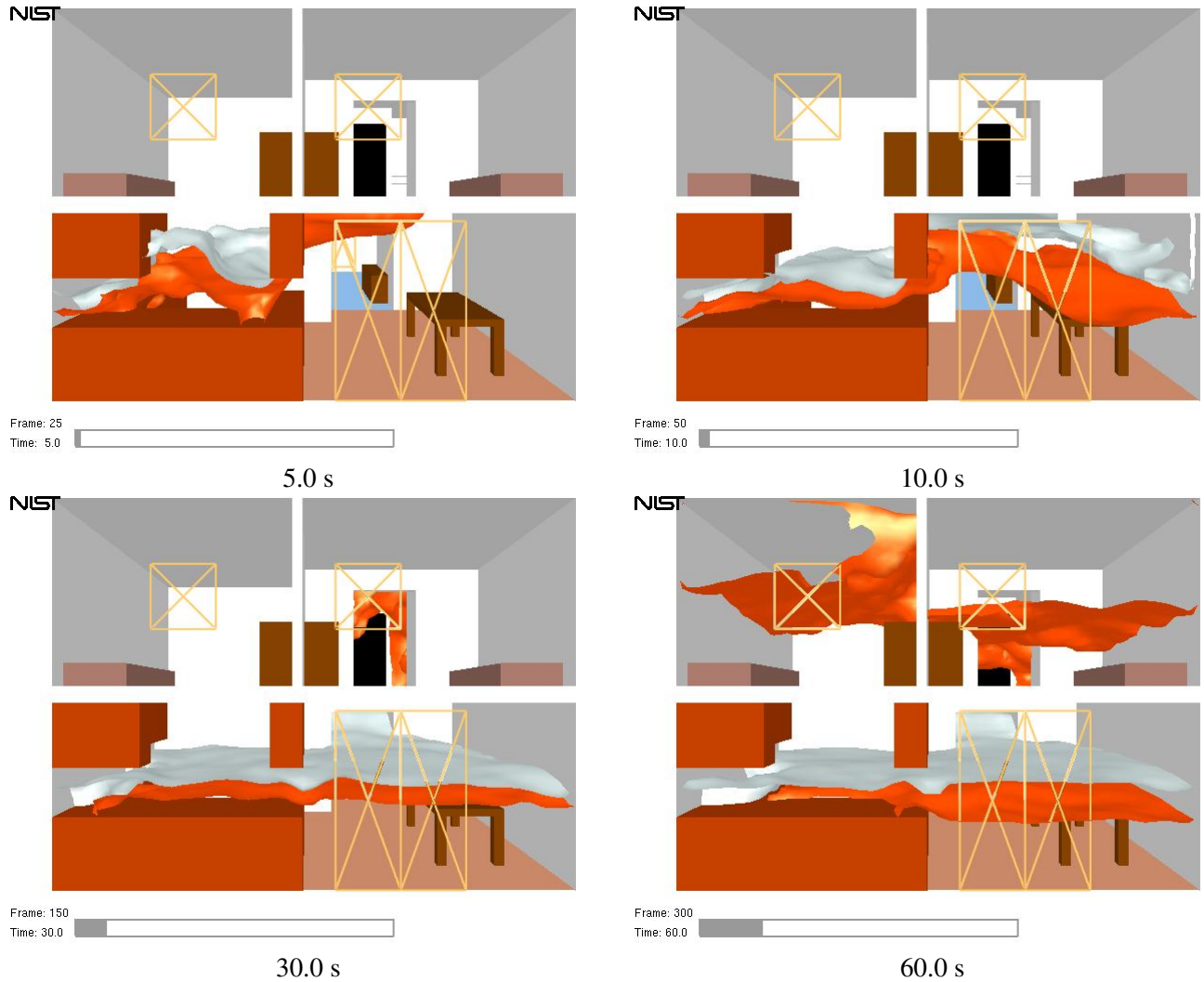
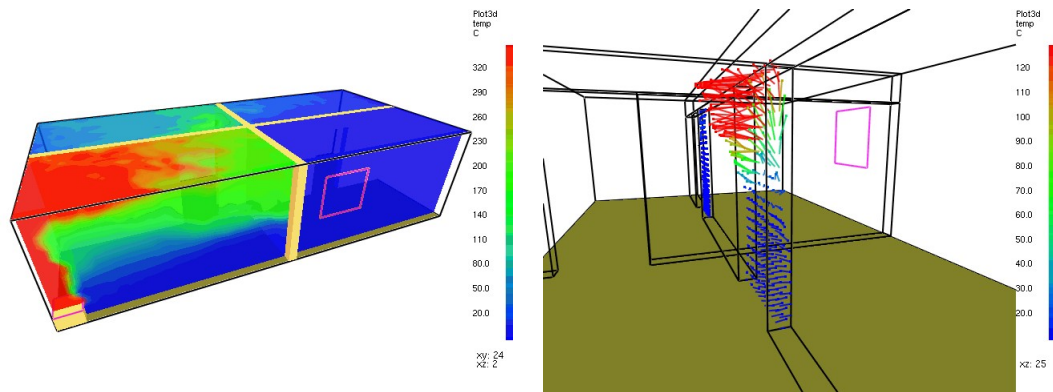


Figure 3.10:

Isosurface file snapshots of temperature levels. The orange surface is drawn where the air/smoke temperature is 30 °C and the white surface is drawn where the air/smoke temperature is 100 °C. These snapshots were generated by adding “&ISO F QUANTITY='TEMPERATURE',VALUE (1)=30.0,VALUE (2)=100.0 /” to the FDS input file.





a) shaded 2D temperature contour plots in a vertical plane through the fire and a horizontal plane below the ceiling

b) shaded temperature vector plot in a vertical plane through the doorway. The “a” key may be depressed to alter the vector sizes. The “s” key may be depressed to alter the number of vectors displayed.

Figure 3.11: Plot3D contour and vector plot examples.

### 3.7 Static Data - Plot3D Files

Data stored in Plot3D files use a format developed by NASA [18] and are used by many CFD programs for representing simulation results. Plot3D files store five data values at each grid cell. FDS uses Plot3D files to store temperature, three components of velocity (U, V, W) and heat release rate. Other quantities may be stored if desired.

An FDS simulation will typically create Plot3D files at several specified times throughout the simulation. Plot3D data is visualized in three ways: as 2D contours, vector plots and isosurfaces. Figure 3.11a shows an example of a 2D Plot3D contour. Vector plots may be viewed if one or more of the U,V and W velocity components are stored in the Plot3D file. The vector length and direction show the direction and relative speed of the fluid flow. The vector colors show a scalar fluid quantity such as temperature. Figure 3.11b shows vectors in a doorway. The vector lengths may be adjusted by depressing the “a” key. Note the hot flow (red color) leaving the fire room in the upper part of the door and the cool flow (blue color) entering fire room in the lower part of the door. Figure 3.12 gives an example of isosurfaces for two different temperature levels. Plot3D data are stored in files with extension .qt.

### 3.8 Zone Fire Modeling Data - PLT Files

The zone fire model CFAST[19] creates files that Smokeview may use to visualize the case being simulated. Smokeview visualizes the geometric layout of the scenario. It also visualizes the layer interface heights, upper layer temperature and vent flow. The vent flow is computed internally in Smokeview using the same equations and data (pressure at floor, layer interface heights and lower/upper layer temperatures) as used by CFAST. The vent flow colors represent the temperature of the flow origin, the length vent flow region is proportional to the flow velocity. Plumes are represented as inverted cones with heights determined using the CFAST reported heat release rate and the same correlation that CFAST uses. A Smokeview view of the one room sample case that comes with the CFAST installation is illustrated in Figure 3.13.



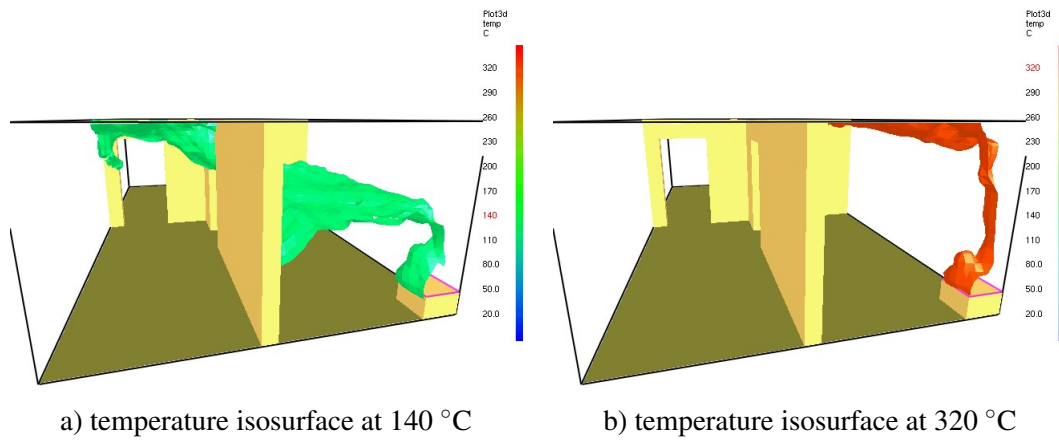


Figure 3.12: Plot3D isocontour example.

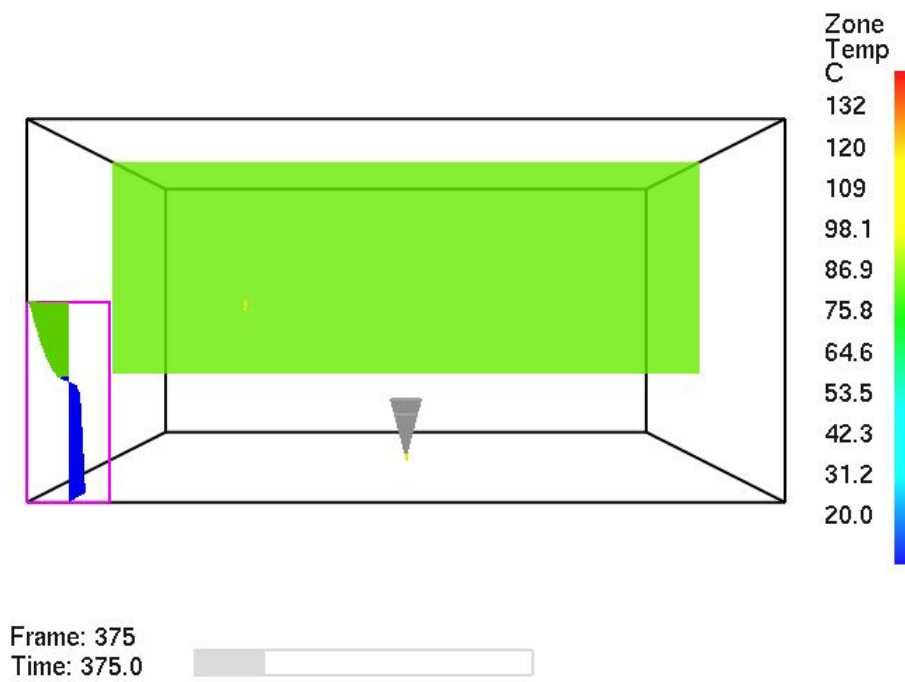


Figure 3.13: CFAST 6.0 Standard case showing upper layer and vent flow at 375 s.



## Chapter 4

# Advanced Features

### 4.1 Manipulating the Scene Automatically - The Touring Option

The touring option allows one to specify arbitrary paths or tours through or around a Smokeview scene. One may then view the scenario from the vantage point of an observer moving along one of these paths. Alternatively, one may observe the motion along all of the paths simultaneously from an external vantage point. These paths may be used to represent the movement of building occupants or first responders during a fire incident, or to highlight key portions or features of a scenario during presentations. A tour may also be used to observe time dependent portions of the scenario such as blockage/vent opening and closings. The default view direction is towards the direction of motion. The view direction and path tension may be changed with the **Advanced Settings** dialog box.

When Smokeview starts up it creates a tour, called the *circle tour* which surrounds the scene. The *circle tour* and a user defined tour are illustrated in Figure 4.1. The circle tour is similar to the Tour menu option found in earlier versions of Smokeview. The user may modify the *circle tour* or define their own tours by using the **Tour** dialog box illustrated in Figure 4.2. The user places several points or keyframes in or around the scene. Smokeview creates a smooth path going through these points.

**Tour Settings** An existing tour may be modified by selecting it from the **Select tour:** listbox found in the Tour dialog box. A new tour may be created by clicking the **Add Tour** button. A newly created tour goes through the middle of the Smokeview scene starting at the front left and finishing at the back right. A tour may also be modified by editing the text entries found in the *local* preference file, casename.ini under the **TOUR** keyword.

The speed traversed along the tour is determined by the time value assigned to each keyframe. If the **Constant Speed** checkbox is checked then these times are determined given the distance between keyframes and the velocity required to traverse the entire path in the specified time as given by the *start time* and *stop time* entries found in the **Advanced Settings** dialog box.

Three different methods for viewing the scene may be selected. To view the scene from the point of view of the selected tour, check the **View From Tour Path** checkbox. To view the scene from a keyframe (to see the effect of editing changes), select the **View From Selected Keyframe** checkbox. Unchecking these boxes returns control of scene movement to the user.

**Keyframe Settings** A keyframe may be selected for editing by either clicking it with the mouse or by using the **Next** or **Previous** buttons. Keyframe positions may then be modified by changing data in the t, X, Y or Z edit boxes. A different view direction may be selected using the **Advanced Settings** dialog box.

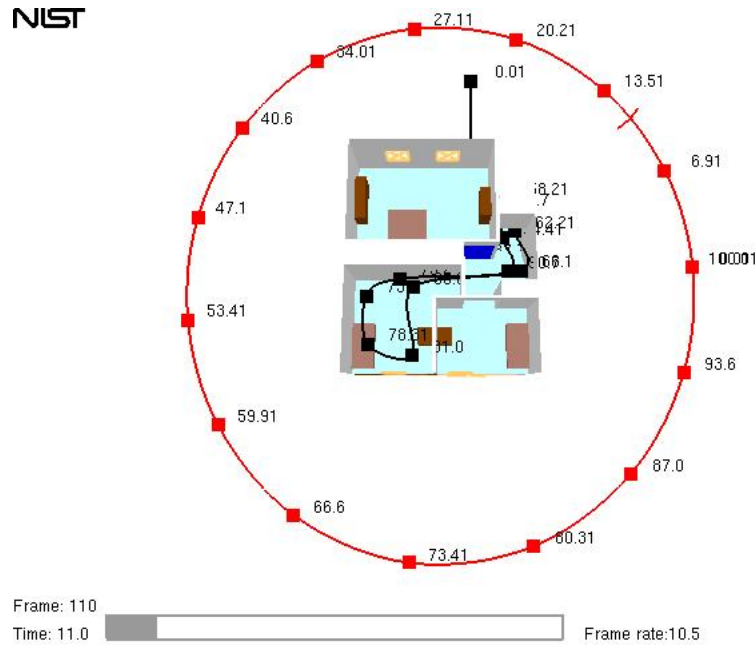


Figure 4.1: Overhead view of townhouse example showing the default *Circle* tour and a user defined tour. The square dots indicate the keyframe locations. Keyframes may be edited using the Touring or Advanced Touring dialog boxes.

A new keyframe is created by clicking the **Add** button. It is formed by averaging the positions and view directions of the current and next keyframes. If the selected keyframe is the last one in the tour then a new keyframe is added beyond the last keyframe.

A keyframe may be deleted by clicking the **Delete** button. There is no **Delete Tour** button. A tour may be deleted by either deleting all of its keyframes or by deleting its entry in the casename.ini file.

**Advanced Settings** The **Advanced Settings** dialog box is only necessary if one wishes to override Smoke-view's choice of view direction (along the direction of motion) or tension settings. This dialog box is opened by clicking the **Advanced Settings** button contained in the **Keyframe Settings** dialog box.

A view direction may be defined at each keyframe by either setting direction angles relative to the path (an azimuth and an elevation angle) or by setting a direction relative to the scene geometry (a cartesian (X,Y,Z) view direction).

Path relative view directions are enabled by default. To define a cartesian view direction, select the **X,Y,Z View** check box and edit the X, Y and Z View edit boxes to change the view location. To define an path relative view direction, uncheck the **X,Y,Z View** check box and edit the Azimuth and Elevation edit boxes. Checking the **View From Selected Keyframe** checkbox allows one to see the effects of the above view changes from the keyframe being edited. To see the effect of a change in one of the keyframe's parameters, uncheck the **View From Selected Keyframe** checkbox and position the tour locator (vertical and horizontal red lines) near the keyframe. The horizontal red line always points in the view direction.

Spline tension settings may also be changed using the **Advanced Settings** dialog box, though normally this is not necessary except when one wishes abrupt rather than smooth path changes. Kochanek-Bartels[16] splines (piecewise cubic Hermite polynomials) are used to represent the tour paths.

The cubic Hermite polynomials for each interval are uniquely specified using a function and a derivative

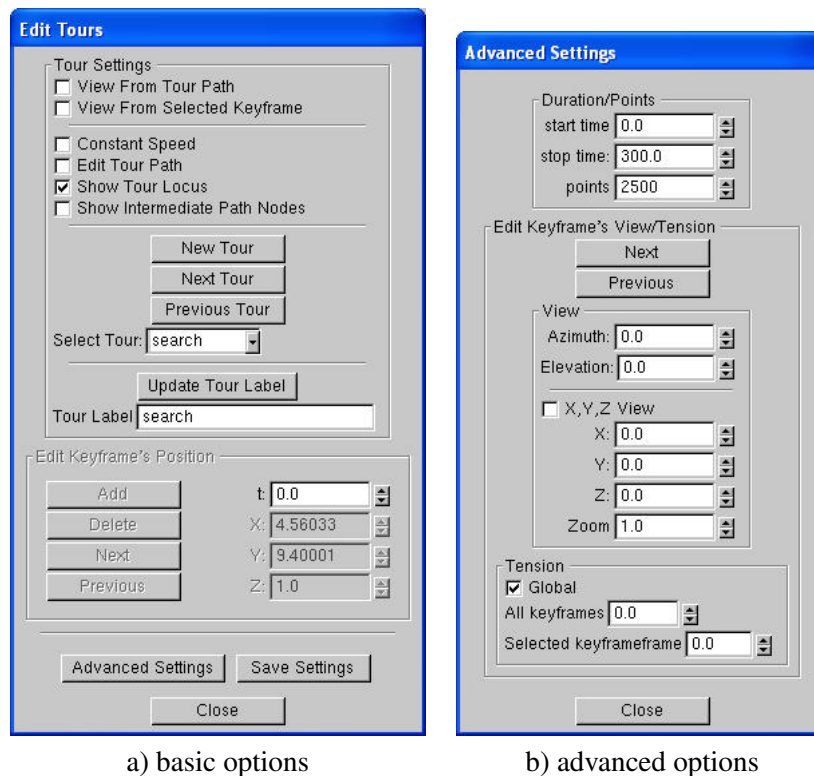


Figure 4.2: The Touring dialog boxes may be used to select tours or keyframes, change the position or view direction at each keyframe and change the tension of the tour path.

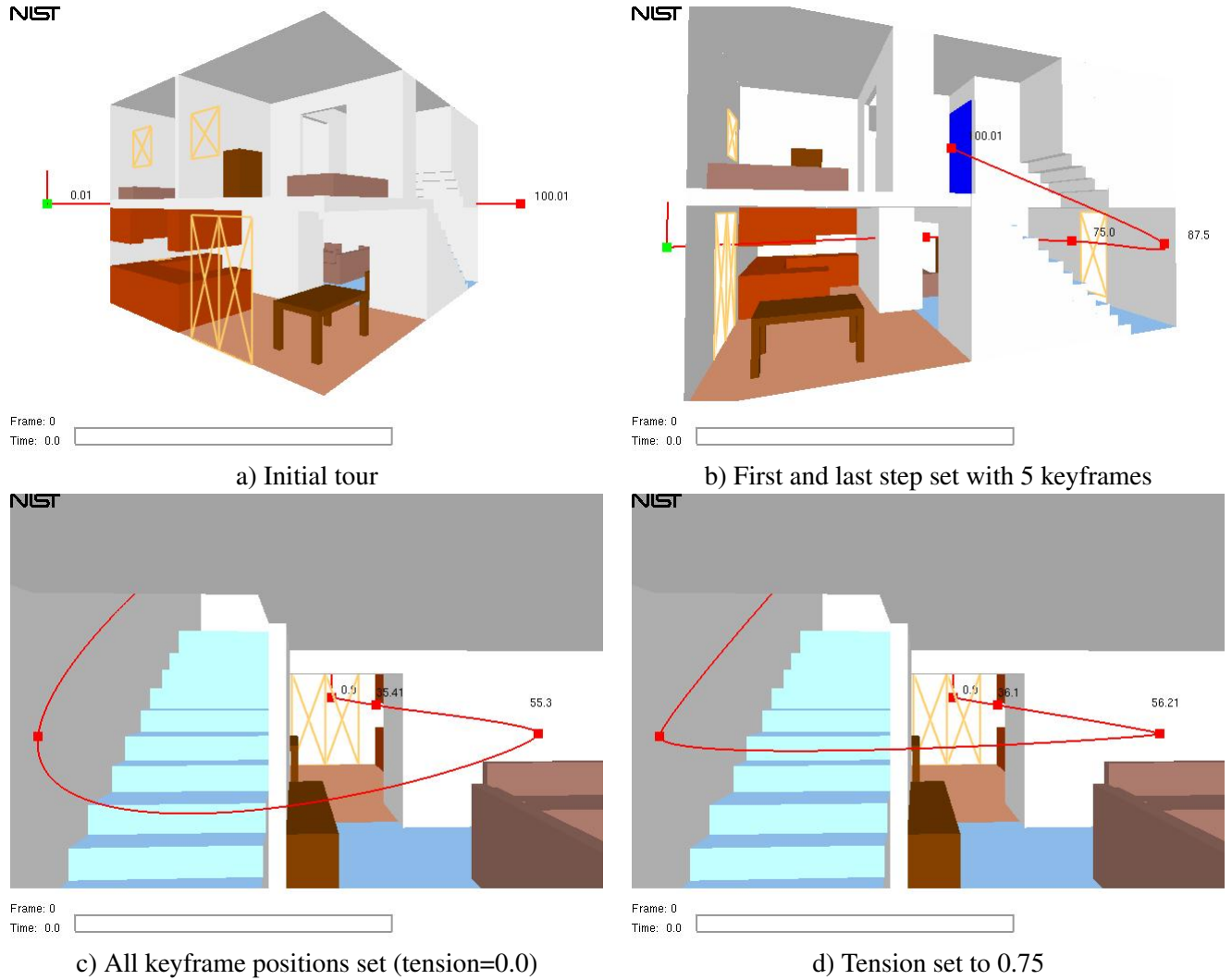


Figure 4.3: Tutorial examples for Tour option.

at both endpoints of the interval (*i.e.* 4 data values). These derivatives are computed in terms of three parameters referred to as *bias*, *continuity* and *tension*. Each of these parameters range from -1 to 1 with a default value of 0. The tension value may be set for all keyframes at once (by checking the **Global** checkbox) or for each keyframe separately. The bias and continuity values are set to zero internally by Smokeview. A tension value of 0 is set by default, a value of 1 results in a linear spline.

**Setting up a tour** The following steps give a simple example of setting up a tour in the townhouse scenario. The tour will begin at the back of the house, go towards the front door and then end at the top of the stairs. These steps are illustrated in Figure 4.3.

1. Start by clicking the `Dialog>Tours...` menu item which opens up the `Edit Tours` dialog box.
2. Click on the `New Tour` button in the Tour dialog box. This creates a tour, illustrated in Figure 4.3a, starting at the front left of the scene and ending at the back right. This tour has two keyframes. The

elevation of each keyframe is halfway between the bottom and top of the scene.

3. Click on the `Edit Tour Path` checkbox. This activates buttons that allows the user to edit the properties of each individual keyframe. Click on the square dot at the back of the townhouse. This is the first keyframe. Click on the square dot at the back of the townhouse. This is the first keyframe. Change the “Z” value to 1.0. Click on the second dot and change its “Z” value to 1.0.
4. Click on the `Add` button, found inside the *Edit Keyframe’s Position* panel, three times. This will add three more key frames to the tour which will be needed so that the path bends up the stairs. You should now have five keyframes.
5. Move the first keyframe at the back of the townhouse near the double door by setting X, Y, Z positions to (3.8,-1.0,1.6). Move the last keyframe to the top of the steps by setting X, Y, Z positions to (6.0,3.6,4.1). The path should now look like Figure 4.3b.
6. Move the second, third and fourth keyframes to positions (4.0,4.0,1.6), (4.0,6.8,1.6) and (6.0,6.8,1.6). The path should now look like Figure 4.3c.
7. Click on the `Advanced Settings` button. Check the `Global` checkbox and set the `All keyframes` edit box to 0.5. This *tightens* up the spline curve reducing the dip near the stairs that occurs with the tension=0.0 setting. The path should now look like Figure 4.3d.
8. Click on the `Save Settings` button to save the results of your editing changes.
9. To see the results of the tour, click on the *View From Tour Path* check box.

The point of view of the observer on this path is towards the direction of motion. Next the view direction will be changed to point to the side while the observer is on the first floor.

1. Click on the `Advanced Settings` button if it is not already open.
2. Uncheck the *View From Tour Path* checkbox in the *tour* dialog box and make sure that the `X, Y, Z View` checkbox is unchecked.
3. Click on the dot representing the first keyframe. Then change *Azimuth* setting to 90 degrees. To see the results of the change, go back and check the *View From Tour Path* checkbox.
4. Uncheck the *View From Tour Path* checkbox again. Now select the second and third keyframes and change their azimuth settings to 90 degrees.

With this second set of changes, the observer will look to the side as they pass through the kitchen and living room. The observer will look straight ahead as they go up the stairs.

## 4.2 Clipping Scenes

It is difficult to view the interior of a scene when modelling complicated geometries. To alleviate this problem, portions of the scene may be hidden or clipped by defining up to six clipping planes. OpenGL draws the scene on one side of a clipping plane but not the other. In general, a clipping plane may have any orientation. Smokeview defines two clipping planes for each of the three coordinate axes. The two x axis clipping planes clip regions with x coordinates smaller than an ‘ $x_{min}$ ’ clipping value and larger than an ‘ $x_{max}$ ’ value. The y axis and z axis clipping planes behave similarly. Clipping plane values are specified

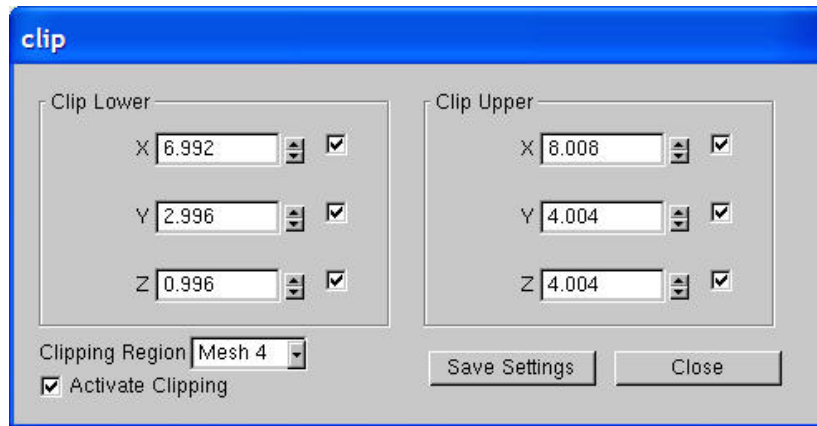
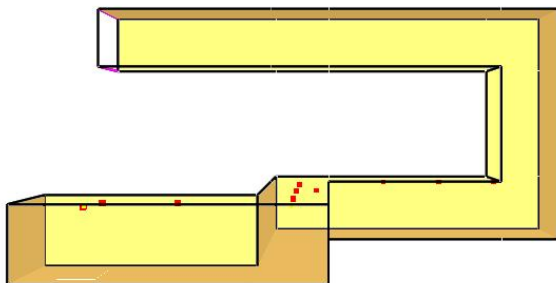
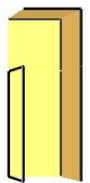


Figure 4.4: Clipping dialog box. Clipping entries result in the view shown in Figure 4.5b. Clipping entries may be changed individually or selected by mesh number.



a) full scene



b) scene clipped to mesh 4

Figure 4.5: Two views of a multi-mesh case. The first view shows the entire scene, the second view shows the scene clipped to mesh 4. The clipping entries producing the second view are shown in Figure 4.4.



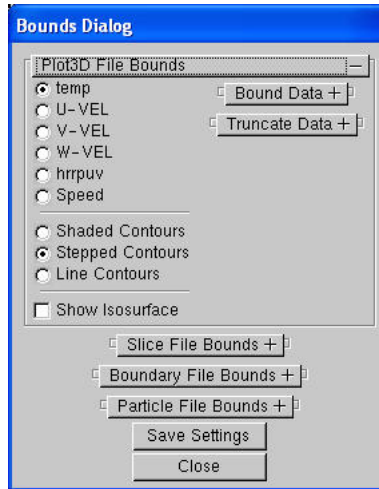


Figure 4.6: File/Bounds dialog box showing PLOT3D file options. Select a variable and a bounds type *checkbox/radio button*, then enter a lower and/or upper bound. Data may be excluded from the plot by selecting a *Truncate* bound. Select type of contour plot to be displayed. Press *Reload...* or *Update* for the new bounds to take effect.

using the clipping dialog box which is opened by selecting the `Dialogs>Clip Geometry` menu item. Figure 4.4 shows this dialog box with all planes active and values selected for mesh 4 of the scene. The corresponding clipped scene is illustrated in Figure 4.5.

### 4.3 Setting Data Bounds

Normally, Smokeview determines data bounds automatically when it loads data. Sometimes, however, it is desirable to override Smokeview's choice. This allows for consistent color shading when displaying several data files simultaneously.

The `Set Bounds...` dialog box is opened from the `Dialogs` menu. Each file type in Figure 4.6 (slice, particle, Plot3D *etc*) has a set of *radio buttons* for selecting the variable type that data bounds are to be applied to. These variable types are determined from the files generated by FDS and are automatically recorded in the `.smv` file. The data bounds are set in a pair of edit boxes. Radio buttons adjacent to the edit boxes determine what type of bounds should be applied. The `Update` and `Reload` buttons are pressed to make the new bounds take effect.

The `Plot3D` and `Slice File` portions of the bounds dialog box have additional controls used to chop or hide data. The settings used in Figure 4.7 were used to generate the ceiling jet visualized in Figure 4.8. Data values less than 140°C are chopped or not drawn in the figure.

Slice file data may be time averaged or smoothed over a user selectable time interval. This option is also implemented from the `Slice File` section of the bounds dialog box (see Figure 4.7).

The **Boundary File** portion of the dialog box has an **ignition** checkbox which allows one to visualize when and where the blockage temperature exceeds its ignition temperature.

The bounds dialog for PLOT3D display allows one to select between three different types of contour plots: shaded, stepped and line contours.

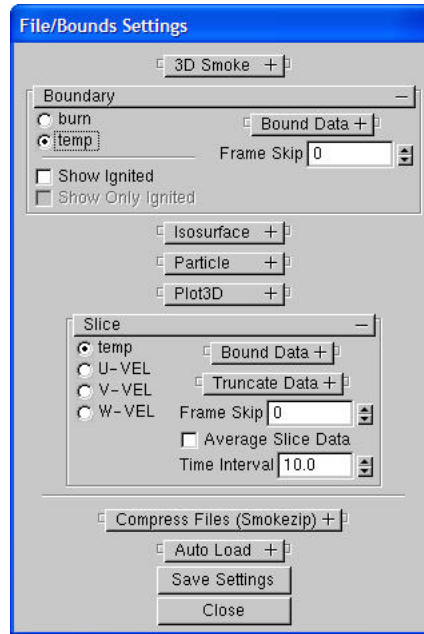


Figure 4.7: File/Bounds dialog box showing slice and boundary file options. Select a variable and a bounds type *check box/radio button*, then enter a lower and/or upper bound. In the slice portion, data may be excluded from the plot by selecting a *Truncate* bound. In the boundary portion, ignited materials may be highlighted if a wall temperature boundary file has been saved. Press *Reload...* or *Update* for the new bounds to take effect.

## 4.4 3D Smoke Options

Figure 4.9 allows one to override Smokeview's choice for several of the 3D smoke parameters. The user may specify the color of the fire and the gray level of the smoke. A gray level of  $n$  where  $n$  ranges from 0 to 7 results in a color of  $(2^n, 2^n, 2^n)$  where the three components represent red, green and blue contributions. The **hrrpuv cutoff** input refers to the heat release rate required at a node before Smokeview will color the node as fire rather than smoke. The **50% flame depth** allows one to specify the transparency or optical thickness of the fire (for visualization purposes only). A small value results in opaquely drawn fire while a large value results in a transparently drawn fire. The **Absorption Parameter** setting refers to how the smoke slices are drawn. The **adjust off-center** setting causes Smokeview to account for non-axis aligned paths. The **adjust off-center + zero at boundary** accounts for off center path lengths and zeros smoke density at boundaries in order to remove graphical artifacts.

## 4.5 Plot3D Viewing Options

Plot3D files are more complicated to visualize than time dependent files such as particle, slice or boundary files. For example, only the transparency and color characteristics of a time file may be changed. With Plot3D files however, many attributes may be changed. One may view 2D contours along the X, Y and/or Z axis of up to six<sup>1</sup> different simulated quantities, view flow vectors and iso or 3D contours. Plot3D file visualization is initiated by selecting the desired entry from the Load/Unload Plot3D sub-menu and

<sup>1</sup>The FDS software stores temperature, three components of velocity (denoted  $u$ ,  $v$  and  $w$ ) and heat release per unit volume. If at least one velocity component is stored in a Plot3D file, then Smokeview adds speed to the Plot3D variable list.

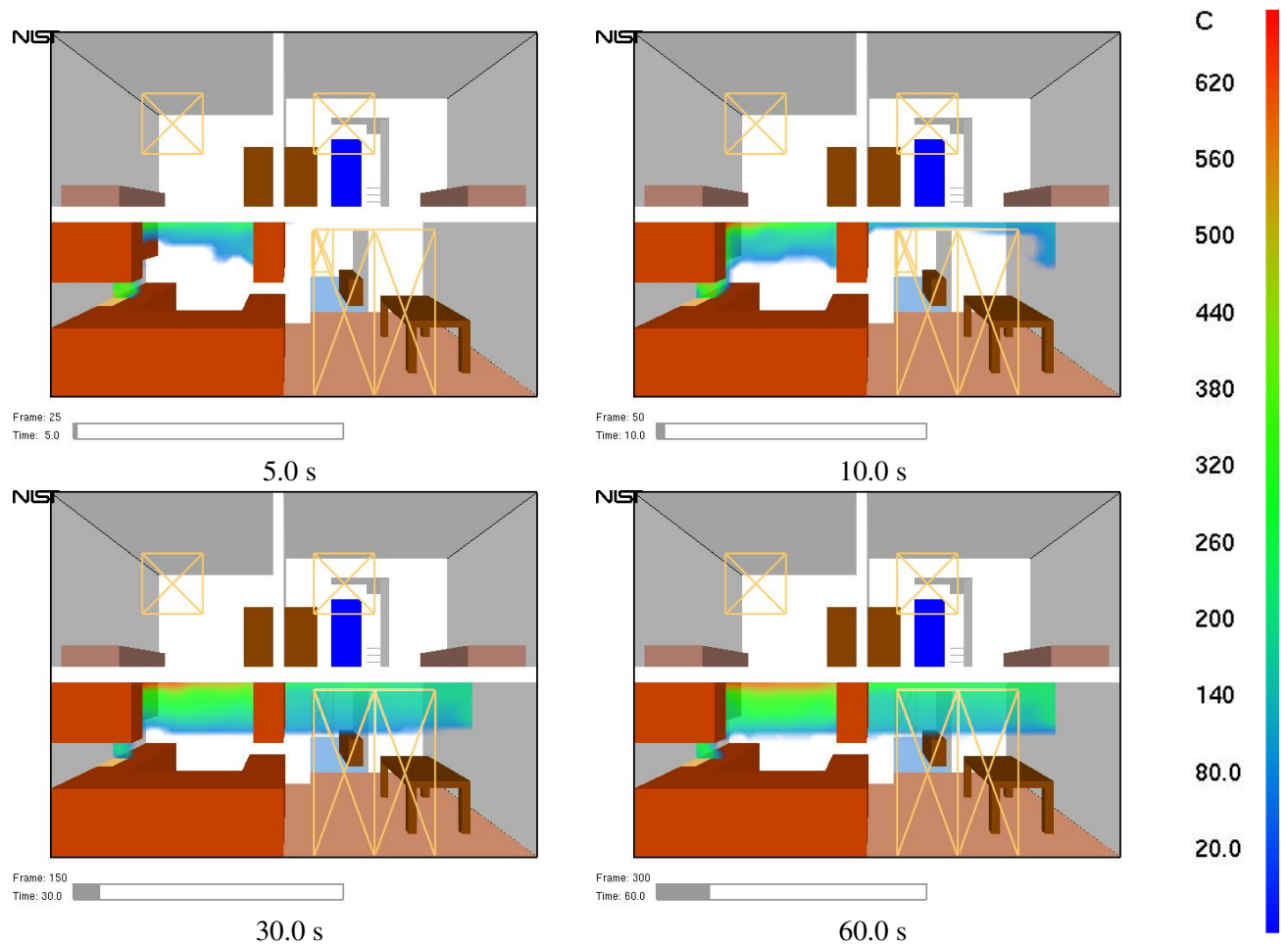


Figure 4.8: Ceiling jet visualization created by *chopping data* below 140 °C using the Bounds Dialog Box as illustrated in Figure 4.7.

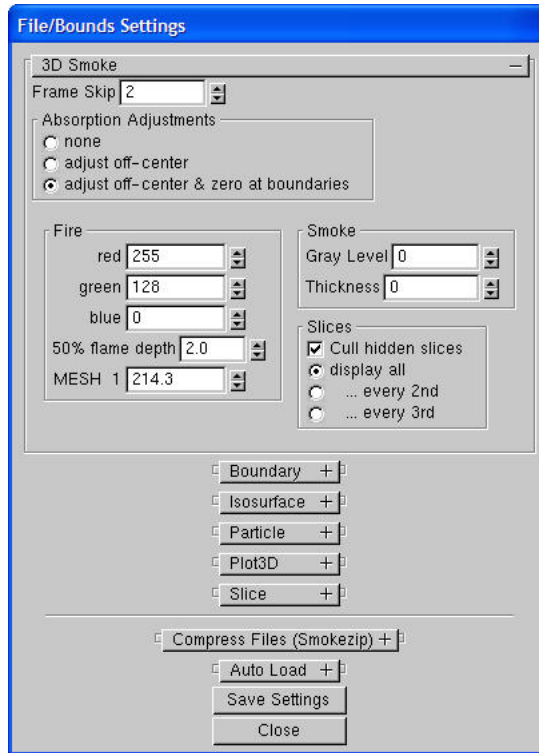


Figure 4.9: Dialog box for setting 3D smoke options.

as with time files one may change color and transparency characteristics.

**2D contours** Smokeview displays a 2D contour slice midway along the Y axis by default when a Plot3D file is first loaded, To step the contour slice up by one grid cell along the Y axis, depress the `space bar`. Similarly to step the contour slice down by one grid cell along the Y axis, depress the “-” key. To view a contour along either the X or Z axis, depress the `x` or `z` keys respectively. Depressing the `x`, `y` or `z` keys while the contour is visible will cause it to be hidden. The Plot3D variable viewed may be changed by either depressing the “p” key or by selecting the `Solution Variable` sub-menu of the `Show/Hide` menu.

**Iso-Contours** Iso-contours also called 3D contours or level surfaces may be viewed by depressing the “i” key or by selecting the `Plot3D 3D Contours` sub-menu of the `Show/Hide` menu.

**Flow vectors** If at least one velocity component is present in the Plot3D file then the “v” key may be depressed in order to view flow vectors. The length and direction of the vector indicates the flow direction and speed. The vector color indicates the value of the currently displayed quantity. A small dot is drawn at the end of the line to indicate flow direction. The vector lengths as drawn may be changed by depressing the “a” key. Vector plots may be very dense when the grid is finely meshed. The “s” key may be depressed in order to skip vectors. For example, all vectors are displayed by default. If the “s” is depressed then every other vector is skipped.



Figure 4.10: Dialog Box for setting miscellaneous Smokeview scene properties

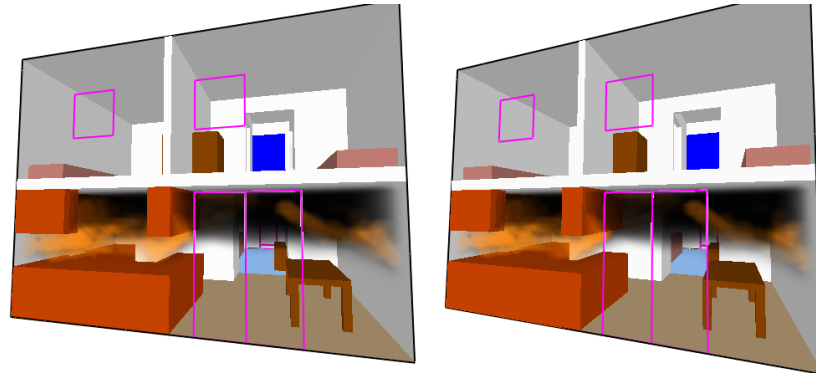


Figure 4.11: Stereo pair view of a townhouse kitchen fire. To aid in viewing the stereo effect, place a finger in front of each image. Relax your eyes allowing your two fingers and stereo pair images to merge into one.

## 4.6 Display Options

### 4.6.1 General

The display dialog box, illustrated in Figure 4.10, allows one to set various options to control the display or *look* of the Smokeview scene. It may be invoked by selecting the `Dialogs>Display` menu item. This dialog box also allows one to show or hide loaded data files.

### 4.6.2 Stereo

Smokeview implements two methods for displaying images in stereo or 3D. Each method creates two versions of the scene, one version for each eye. In one method, the two images are displayed alternately. Specialized shuttered glasses are synched with the monitor to ensure that the left eye sees only the left eye's version of the scene and that the right eye sees only the right eye's version of the scene. To be effective the monitor requires a refresh rate of at least 120 frames per second (60 frames per second for each eye). Flickering is annoyingly noticeable in monitors with slower frame rates. Unfortunately, today's LCD flat panel monitors typically do not have refresh rates faster than 60 to 80 frames per second. This method requires a video card that implements OpenGL *QUAD buffering*. This Smokeview option can only be enabled from the command line by using the `-stereo` option.

In the second method, Smokeview displays the two images side by side, again one image for each eye. With practice, one can merge the images without requiring specialized equipment, especially if the images are small and not separated by a large angle. A trick for seeing the stereo effect is to place a finger from



Figure 4.12: Dialog box for activating the stereo view option.

each hand in the center of each picture. Then relax your eyes, trying to *merge* your two fingers together.

Figure 4.11 shows a pair of images of a townhouse kitchen fire. Figure 4.12 shows the dialog box used to activate this option. If the *-stereo* command line option was used and the video card supports shuttered stereo display then the *shuttered checkbox* in the dialog box will be enabled.

## 4.7 Texture Maps

Texture mapping is a technique used by Smokeview to make a scene appear more realistic by pasting images onto obstructions or vents. For example, to apply a wood paneling image to a wall, add the keywords `TEXTURE_MAP='paneling.jpg'`, `TEXTURE_WIDTH=1.`, `TEXTURE_HEIGHT=2.` to the `&SURF` line where **paneling.jpg** is the JPEG file containing the texture map (SGI users should use RGB files instead of JPEG) and **TEXTURE\_WIDTH** and **TEXTURE\_HEIGHT** are the characteristic dimensions of the texture map in meters. Note that the image will not appear when Smokeview first starts up. The user must select the texture maps using the Show/Hide menu.

One can create texture maps using a digital camera or obtain them commercially. The maps should be *seamless* so that no breaks or seams appear when the maps are tiled on a blockage or vent. This is important, because Smokeview replicates the image as often as necessary to cover the blockage or vent.

When the texture does have a pattern, for example windows or bricks, the keyword `TEXTURE_ORIGIN` may be used to specify where the pattern should begin. For example,

```
&OBST XB=1.0,2.0,3.0,4.0,5.0,7.0,SURF_ID='wood paneling',
      TEXTURE_ORIGIN=1.0,3.0,5.0 /
```

will apply paneling to an obstruction whose dimensions are 1 m by 1 m by 2 m, such that the image of the paneling will be positioned at the point (1.0,3.0,5.0). The default value of `TEXTURE_ORIGIN` is (0,0,0), and the global default can be changed by added a `TEXTURE_ORIGIN` statement to the `MISC` line.

Figure 4.13 shows a simple application of a texture applied to two different blockages and a vent. The same jpeg file was used in two different `&SURF` lines so that the texture could be stretched by differing amounts (using the `TEXTURE_WIDTH` parameter.) The FDS data file used to create this Figure follows.

```
&HEAD CHID='sillytexture',TITLE='Silly Test Case' /
&MISC TEXTURE_ORIGIN=0.1,0.1,0.1 /

&GRID IBAR=20,JBAR=20,KBAR=20 /
&PDIM XBAR=1.0,YBAR=1.0,ZBAR=1.0 /
&TIME TWFIN=0. /
```



Figure 4.13: Texture map example. The same texture was applied to two different blockages and a vent (with different widths) by assigning different TEXTURE\_WIDTH parameters in the input file.

```
&SURF ID      = 'TEXTURE 1'
  TEXTURE_MAP= 'nistleft.jpg'
  TEXTURE_WIDTH=0.6
  TEXTURE_HEIGHT=.20
  rgb=0.0,0.0,1.0
/

&SURF ID      = 'TEXTURE 2'
  TEXTURE_MAP= 'nistleft.jpg'
  TEXTURE_WIDTH=0.4
  TEXTURE_HEIGHT=.20
  rgb=1.0,0.0,1.0
/

&OBST XB=0.1,0.30, 0.1,0.7, 0.1,0.3, SURF_ID='TEXTURE 1' /
&OBST XB=0.5,0.9, 0.3,0.7, 0.1,0.5 SURF_ID='TEXTURE 2'
  rgb=1.0,0.0,0.0 TEXTURE_ORIGIN=0.5,0.3,0.1 /

&VENT XB=0.00,0.00,0.2,0.8,0.20,0.40,SURF_ID='TEXTURE 1'
  rgb=0.0,1.0,0.0 TEXTURE_ORIGIN=0.0,0.2,0.2 /
&VENT XB=0.30,0.90,0.0,0.0,0.30,0.50,SURF_ID='TEXTURE 1'
  TEXTURE_ORIGIN=0.90,0.0,0.3 /
```

## 4.8 Using Smokeview to Debug FDS Input Files

One of the most difficult tasks in setting up an FDS input file is defining the geometry (blockages, vent locations *etc*) properly. Smokeview may be used to debug FDS input files by making short runs and observing whether blockages, vents and other geometric features of a run are located correctly. Blockages may then be created or changed. This is done with the *Edit Blockage* dialog box called from the Option menu.

The following is a general procedure for identifying problems in FDS input files. Assume that the FDS input data file is named `testcase1.data`.

1. Set the stop time to 0.0 using `TWFIN=0.0` on the `&TIME` line. This causes FDS to read the input file and create a `.smv` file without performing lengthy startup calculations.
2. After the input file has been modified, run the FDS model (for details see the FDS User's Guide[17])  
FDS creates a file named `testcase1.smv` containing information that Smokeview uses to visualize data.
3. To visualize the input file, open `testcase1.smv` with Smokeview by either typing `smokeview testcase1` at a command shell prompt or if on the PC by double-clicking the file `testcase1.smv`.
4. Make corrections if necessary. Use the `COLOR` or `RGB` option<sup>2</sup> of the `OBST` keyword to more easily identify blockages to be edited. For example, to change a blockage's color to red use:

```
&OBST XB=0.0,1.0,0.0,1.0,0.0,1.0 COLOR='RED' /
```

or

```
&OBST XB=0.0,1.0,0.0,1.0,0.0,1.0 RGB=1.0,0.0,0.0 /
```

Save `testcase1.data` file and go back to step 2.

5. If corrections are unnecessary, then change the `TWFIN` keyword back to the desired final simulation time, remove any unnecessary FDS `COLOR` keywords and run the case.

**Examining and/or Editing Blockages** Blockages may be examined or changed by selecting the menu item Dialogs/Edit Geometry

which opens up the dialog box illustrated in Figure 4.14. Text edit boxes allow one to define comments (any text to the right of the closing `/` on an `&OBST` line and pull down menus for linking surfaces (`&SURF` lines) to obstructions. The surface choices are obtained from the input file or the database file if one is specified in the input file.

Note, the blockage editor does not work when clipping planes are active.

The user may associate one, three or six surfaces with an obstruction. If the `3 wall` radio button is selected then materials for the ceiling (upper surface), walls (vertical surfaces) and floor (lower surface) may be specified. The `color` check box allows the inputted or default color (manilla) to be displayed on blockage surfaces.

Associating unique colors with each surface allows the user to quickly determine whether blockages are defined with the proper surfaces. One can then verify that these modelling elements have been defined and positioned as intended.

---

<sup>2</sup>The FDS `COLOR` keyword may have values: RED, BLUE, GREEN, MAGENTA, CYAN, YELLOW, WHITE or BLACK. The `RGB` keyword uses three floating point inputs ranging from 0.0 to 1.0 to specify the red, green and blue components of color. `COLOR` replaces `BLOCK.COLOR` and `VENT.COLOR`.



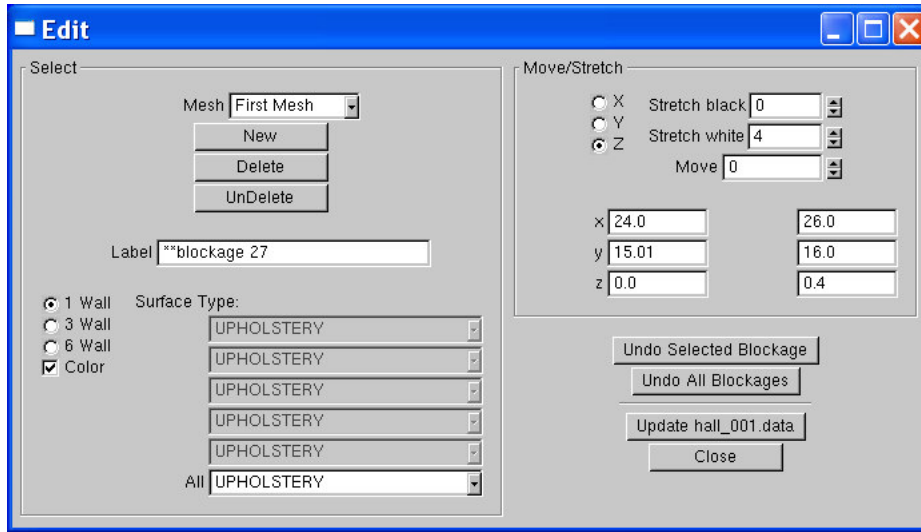


Figure 4.14: Blockage Edit Dialog Box. Click on ‘New’ to create a new blockage or click on an existing blockage in the Smokeview scene to change its size or location. Change the size or location by entering blockage bounds directly or clicking an ‘arrow’ and/or direction buttons. Note, the blockage editor does not work when clipping planes are active.

The blockage editor works on one mesh at a time. For multi-mesh cases, use the Mesh list-box to select the desired mesh or simply click on a blockage in the mesh to switch to that mesh. The pull-down menu is necessary for meshes without blockages.

Position coordinates entered into the text boxes are *snapped* to the nearest grid line. Materials appearing under Surface Type come from those materials defined in either the input data file or the database file. Either one, three or six wall properties can be entered by selecting the appropriate radio button.

## 4.9 Making Movies

A movie can be made of a Smokeview animation by converting the visualized scene into a series of PNG or JPEG files, one file for each time step and then combining the individual images together into one movie file. More specifically:

1. Set up Smokeview by orienting the scene and loading the desired data files.
2. Select the `Options/Render` menu and pick the desired frame skip value. The more frames you include in the animation, the smoother it will look. Of course more frames result in larger file sizes. Choose fewer frames if the movie is to appear on a web site.
3. Use a program such as the Antechinus Media Editor available at <http://www.c-point.com> to assemble the JPEGs or PNGs rendered in the previous step into a movie file.

The default Smokeview image size is  $640 \times 480$ . This size is fine if the movie is to appear in a presentation located on a local hard disk. If the movie is to be placed on a web site then care needs to be taken to insure that the generated movie file is a reasonable size. Two suggestions are to reduce the image size to  $320 \times 240$  or smaller by modifying the `WINDOWWIDTH` and `WINDOWHEIGHT` smokeview.ini keywords and to reduce the number of frames to 300 or less by skipping intermediate frames *via* the `Options/Render` menu.

Sometimes when copying or *capturing* a Smokeview scene it is desirable, or even necessary, to have a margin around the scene. This is because the capturing system does not include the entire scene but itself captures an indented portion of the scene. To indent the scene, either press the “h” key or select the `Option>Viewpoint>Offset Window` menu item. The default indentation is 45 pixels. This may be changed by adding/editing the `WINDOW_OFFSET` keyword in the `smokeview.ini` file.

## 4.10 Annotating the Scene

Smokeview rendered JPEG or PNG image files may be annotated with any package that edits these types of files. Tick marks and label annotation can be placed within the 3D scene and moved around as the scene is moved. This would be difficult to duplicate with an external image editor. In addition, axis labels are displayed, if desired, by selecting `Show/Hide > Labels > Axis labels`.

The current release of FDS places tick marks and labels documenting the scene dimensions. To replace or customize these annotations add the `TICK` keyword to a `.smv` file using the following format:

```
TICKS
xb yb zb xe ye ze nticks
ticklength tickdir r g b tickwidth
```

where `xb`, `yb`, and `zb` are the x, y and z coordinates of the first tick; `xe`, `ye` and `ze` are the x, y and z coordinates of the last tick and `nticks` is the number of ticks. The coordinate dimensions are in physical units, the same units used to set up the FDS geometry. The parameter `ticklength` specifies the length of the tick in physical units. The parameter `tickdir` specifies the tick direction. For example 1(-1) places ticks in the positive(negative) x direction. Similarly, 2(-2) and 3(-3) place ticks in the positive(negative) y and positive(negative) z directions.

The color parameters `r`, `g` and `b` are the red, green and blue components of the tick color each ranging from 0.0 to 1.0. The foreground color (white by default) may be set by setting any or all of the `r`, `g` and `b` components to a negative number. The `tickwidth` parameter specifies tick width in pixels. Fractional widths may be specified.

The `LABEL` keyword allows a text string to be added within a Smokeview scene. The label color and start and stop appearance time may also be specified. The format is given by

```
LABELS
x y z r g b tstart tstop
label
```

where (`x`, `y`, `z`) is the label location in cartesian coordinates and `r`, `g`, `b` are the red, green and blue color components ranging from 0.0 to 1.0. Again, if a negative value is specified then the foreground color will be used instead (white is the default). The parameters, `tstart` and `tstop` indicate the time interval when the label is visible. The text string is specified on the next line (`label`).

Figure 4.15 shows how the `LABEL` and `TICKS` keywords can be used together to create a *ruler* with major and minor tick marks. These ticks and labels were created using:

```
TICKS
0.0 0.0 0.0 8.0 0.0 0.0 5
0.5 -2.0 -1. -1.0 -1.0 4.0
TICKS
1.0 0.0 0.0 9.0 0.0 0.0 5
0.25 -2.0 -1. -1.0 -1.0 4.0
```

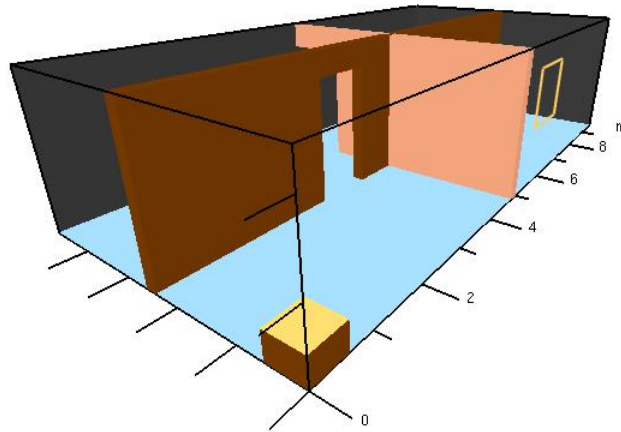


Figure 4.15: Annotation example using the TICKS and LABEL keyword.

```

TICKS
0.0 0.0 0.0 0.0 0.0 2.0 3
0.5 -1.0 -1. -1.0 -1.0 4.0
TICKS
0.0 0.0 0.0 0.0 4.0 0.0 5
0.5 -1.0 -1. -1.0 -1.0 4.0
LABEL
0.0 -0.6 0.0 -1.0 0.0 0.0 0.0 20.0
0
LABEL
2.0 -0.6 0.0 -1.0 0.0 0.0 0.0 20.0
2
LABEL
4.0 -0.6 0.0 -1.0 0.0 0.0 0.0 20.0
4
LABEL
6.0 -0.6 0.0 -1.0 0.0 0.0 0.0 20.0
6
LABEL
8.0 -0.6 0.0 -1.0 0.0 0.0 0.0 20.0
8
LABEL
9.5 -0.6 0.0 -1.0 0.0 0.0 0.0 20.0
m

```

## 4.11 Creating Custom Devices

Smokeview draws devices such as heat and smoke detectors, sprinklers, sensors *etc.* using instructions contained in a data file. These instructions correspond to OpenGL library calls, the same type of calls Smokeview uses to visualize FDS cases. Smokeview then acts as an OpenGL interpreter executing OpenGL

```

DEVICEDEF
    //  comments follow the double slash

    //  blank lines ^^^^^ are permitted
    //  any line not begging with a double "slash"
    //  is part of the devices definition.
device_name // name or type of device
// comments and blank lines

// are permitted anywhere
args command1 args command2 ... // arguments followed by command name
args command1 args command2 ... // arguments and commands may be split
                                // across lines
NEWFRAME    // beginning of next frame
args command args command ...
....

```

Figure 4.16: Device Object file format

commands as specified in the device definition file. Efficiency is obtained by compiling these instructions into display lists, terminology for an OpenGL construct for storing and efficiently drawing collections of OpenGL *commands*.

This section describes how to create new devices. Though all of the examples are given for devices, the intent of this procedure is to be more general allowing Smokeview to draw other types of objects such as people walking.

There are two types of instructions, instructions for drawing simple geometric objects such as cubes, disks, spheres *etc.* and instructions for manipulating these objects through scaling, rotation and translation. Collectively these instructions specify the type, location and orientation of objects used to represent devices. The important feature of this process is that new devices may be designed and drawn without the need to modify Smokeview.

#### 4.11.1 Device File Format

The format for an object definition file is given in Figure 4.16. Each device definition consists of one or two frames. Devices such as thermocouples which do not activate use just one frame. Other devices such as sprinklers or smoke detectors which do activate use two frames, the first for normal conditions and the second for when the device has activated.

Each definition begins with the `DEVICEDEF` keyword. The next non-comment line contains a label naming the device. Succeeding lines contain one or more commands, each pre-pended by zero or more numerical arguments. New frames begin with the `NEWFRAME` keyword. Blank lines may occur anywhere. Definitions may be documented by adding comments following the `'''` characters.

Some examples of command/argument pairs are `d drawsphere` for drawing a sphere of diameter `d` or `x y z translate` for translating an object by  $(x,y,z)$ . Transformation commands are cumulative, each command builds on the effects of the previous command. The `push` and `pop` then may be used to isolate these effects by saving and restoring the geometric state. More details may be found in the next two sections.

```

DEVICEDEF
sensor
1.0 1.0 0.0 setcolor
0.038 drawsphere

```

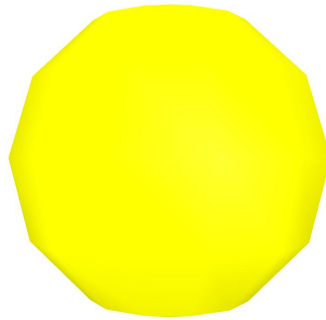


Figure 4.17: Instructions for drawing a sensor along with the corresponding Smokeview view.

A simple example of a definition used to draw a sensor along with the corresponding Smokeview view is given in Figure 4.17. The definition uses just one frame. A sphere is drawn with color yellow and diameter 0.38 m. Push and pop commands are not necessary because there is only one object and no transformations are used.

Figure 4.18 is more complicated. It shows a definition for an up-right sprinkler along with a corresponding Smokeview view. The definition uses two frames. The first frame represents the sprinkler's inactive state, the second frame represents the active state (commands after the `NEWFRAME` keyword). This definition uses cubes, a truncated cone, spheres, disks and a notched plate. The cube is stretched and rotated using the `scale` and `rotate` commands forming the two sloped 'bars'. The other parts are moved into their proper position using the `translate` command. The push and pop commands are used to isolate the effects of transformation commands for one part when drawing another.

As with preference (.ini) files, Smokeview looks for device definitions in three places. First, in a file named `devices.svo`, in the directory containing the Smokeview program. On the PC, this is `C:\nist\fds\`. Next, in a file named `devices.svo`, in the directory containing the case being viewed and finally in a file named `casename.svo` where `casename` is the name of the case.

The global file, `devices.svo`, contains definitions for a sprinkler (upright and pendent), heat detector, smoke detector, sensor and nozzle. Figure 4.19 gives Smokeview views for these devices. Active views are shown when they are defined. The object's origin is identified by the three intersecting lines. The origin is placed where FDS records data for these devices.

## Heat detector Instructions

```
DEVICEDEF
heat_detector          // label, name of device

// The heat detector has three parts
//   a disk, a truncated disk and a sphere.
//   The sphere changes color when activated.

0.8 0.8 0.8 setcolor // set color to off white
push 0.0 0.0 -0.02 translate 0.127 0.04 drawdisk pop
push 0.0 0.0 -0.04 translate 0.06 0.08 0.02 drawtrunccone pop
0.0 1.0 0.0 setcolor
push 0.0 0.0 -0.03 translate 0.04 drawsphere pop
// push and pop are not necessary in the last line
//   of a frame. Its a good idea though, to prevent
//   problems if parts are added later.
NEWFRAME // beginning of activated definition
0.8 0.8 0.8 setcolor
push 0.0 0.0 -0.02 translate 0.127 0.04 drawdisk pop
push 0.0 0.0 -0.04 translate 0.06 0.08 0.02 drawtrunccone pop
1.0 0.0 0.0 setcolor
push 0.0 0.0 -0.03 translate 0.04 drawsphere pop
```

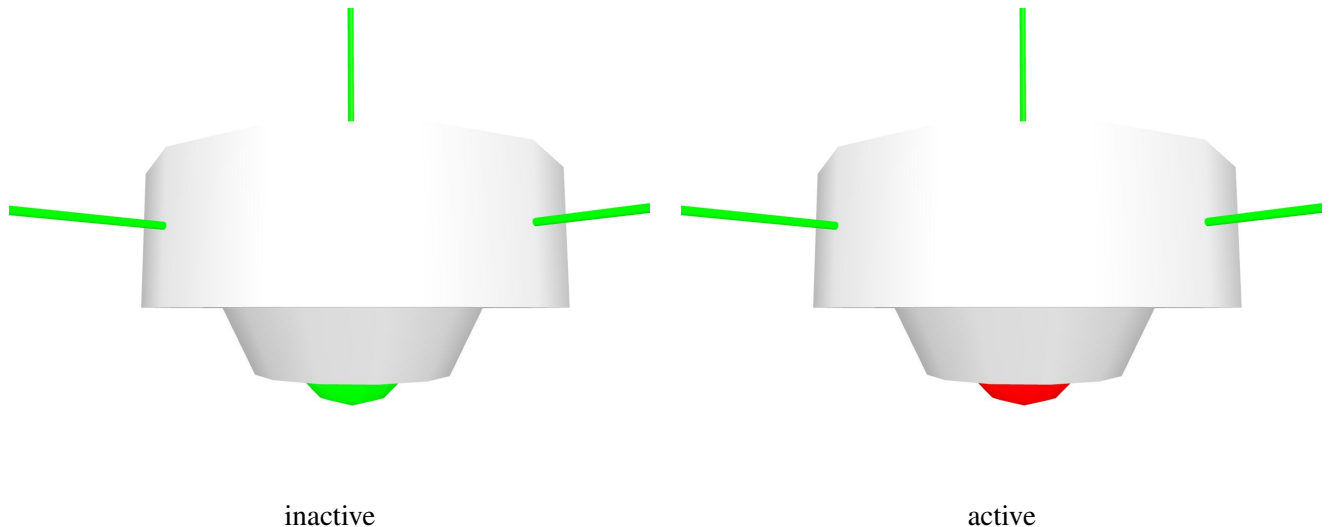


Figure 4.18: Instructions for drawing an inactive and active heat detector along with the corresponding Smokeview view

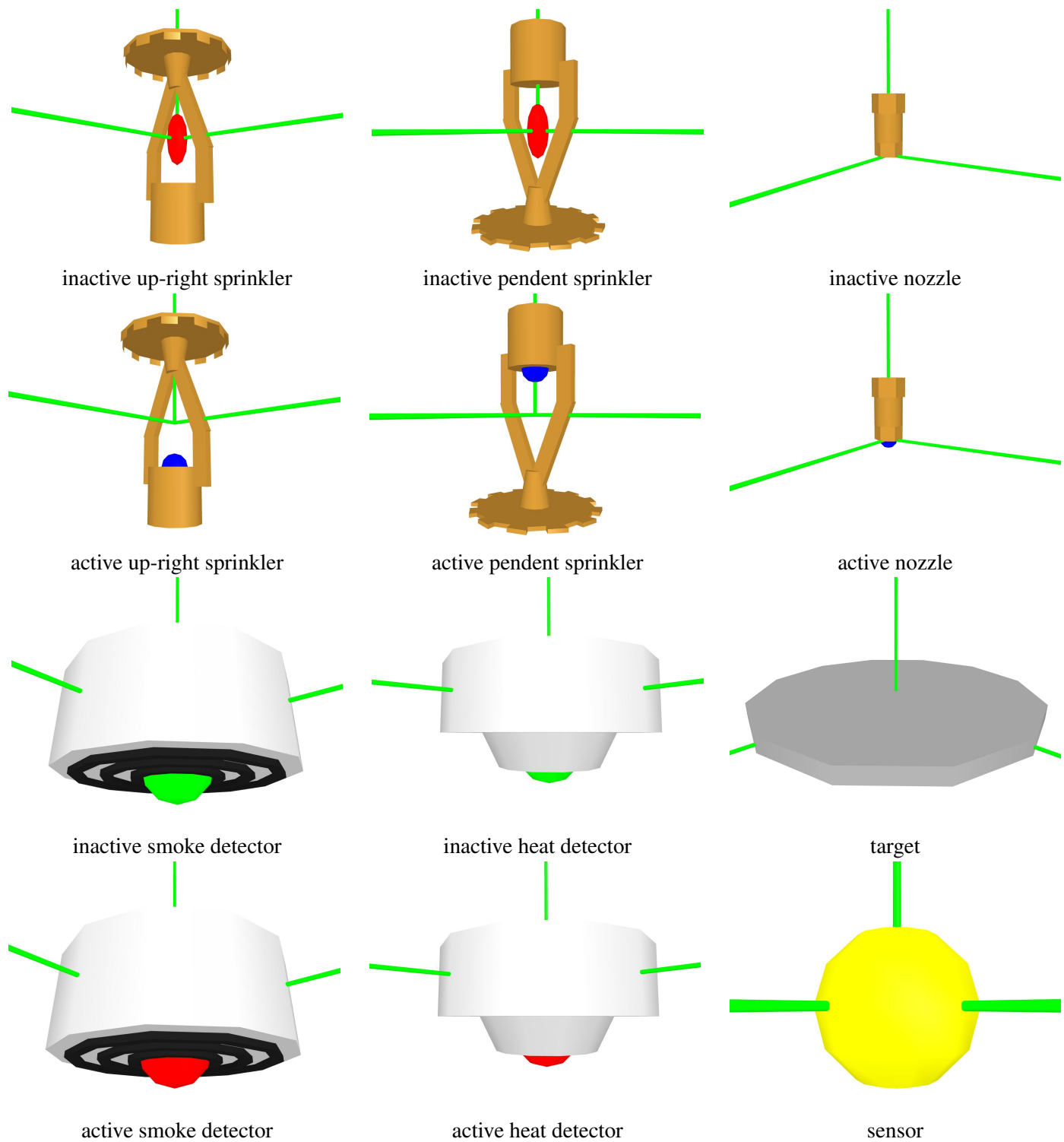


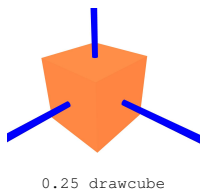
Figure 4.19: Smokeview view of devices defined in the global devices.svo file. The device origin occurs at the intersection of the thin (green) lines, where data is recorded in FDS.

### 4.11.2 Geometric Objects

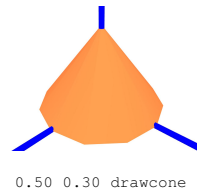
Each command used to draw elementary geometric objects consists of one or more arguments followed by the command *e.g.* `0.3 drawsphere`. Some portion of the object is designated as the origin, *i.e.* with coordinate  $(0,0,0)$ . The location of the origin is needed when assembling geometric objects together to form more complex objects. The transformation commands discussed in the next section are used to perform this assembly, specifying exactly where and with what orientation the geometric objects should be placed.

**DRAWCIRCLE** The command, `d drawcircle`, draws a circle with diameter *d*. The origin is located at the center of the circle.

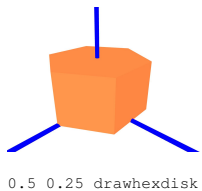
**DRAWCONE** The command, `d h drawcone`, draws a right circular cone where *d* is the diameter of the base and *h* is the height. The origin is located at the center of the cone's base.



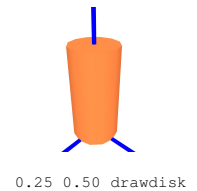
**DRAWCUBE** The command, `s drawcube`, draws a cube where *s* is the length of the side. The origin is located at the center of the cube. An oblong box, a box with different length sides, may be drawn by using `scale` along with `drawcube`. For example, `1.0 2.0 4.0 scale 1.0 drawcube` creates a box with dimensions  $1 \times 2 \times 4$ .



**DRAWDISK** The command, `d h drawdisk`, draws a circular disk with diameter *d* and height *h*. The origin is located at the center of the disk's base.

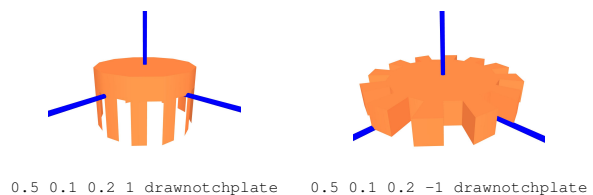


**DRAWHEXDISK** The command, `d h drawhexdisk`, draws a hexagonal disk with diameter *d* and height *h*. The origin is located at the center of the hexagon's base.



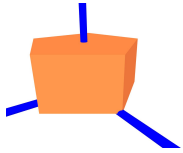
**DRAWLINE** The command, `x1 y1 z1 x2 y2 z2 drawline`, draws a line between the points  $(x_1, y_1, z_1)$  and  $(x_2, y_2, z_2)$ .

**DRAWNOTCHPLATE** The command, `d h nh dir drawnotchplate`, draws a notched plate. This object is used to represent a portion of a sprinkler where *d* is the plate diameter, *h* is the plate height (not including notches), *nh* is the height of the notches and *dir* indicates the notch orientation (1 for vertical, -1 for horizontal). The origin is located at the center of the plate's base.



**DRAWPOINT** The command, `drawpoint`, draws a point (small square). The command, `s setpointsize` may be used to change the size of the point. The default size is 1.0.

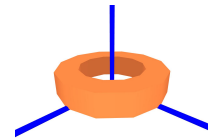




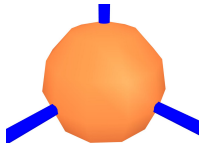
5 0.35 0.15 drawpolydisk

**DRAWPOLYDISK** The command, `n d h drawpolydisk`, draws an  $n$ -sided polygonal disk with diameter  $d$  and height  $h$ . The origin is located at the center of the polygonal disk's base. The example to the left is a pentagonal disk.

**DRAWRING** The command, `di do h drawring`, draws a ring where  $di$  and  $do$  are the inner and outer ring diameters and  $h$  is the height of the ring. The origin is located at the center of the ring's base.



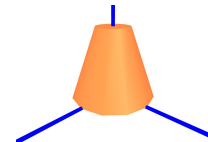
0.3 0.5 0.1 drawring



0.25 drawsphere

**DRAWSPHERE** The command, `d drawsphere`, draws a sphere with diameter  $d$ . The origin is located at the center of the sphere. As with an oblong box, an ellipsoid may be drawn by using `scale` along with `drawsphere`. For example, `1.0 2.0 4.0 scale 1.0 drawsphere` creates an ellipsoid with semi-major axes of length 1, 2 and 4. This is how the sprinkler *bulb* in Figure 4.18a is drawn.

**DRAWTRUNCCCONE** The command, `d1 d2 h drawtrunccone`, draws a right circular truncated cone where  $d1$  is the diameter of the base,  $d2$  is the diameter of the truncated portion of the cone and  $h$  is the height or distance between the lower and upper portions of the truncated cone. The origin is located at the center of the truncated cone's base.



0.5 0.2 0.4 drawtrunccone

### 4.11.3 Transformations

As with geometric commands, transformation commands consist of one or more arguments followed by the command (except for `push` and `pop` which do not require any arguments). Transformation commands are used to change the location and orientation of drawn objects, to save and restore the geometric state and to set attributes such as point size, line width or object color. The `rotate` and `translate` commands change the origin (`translate`) or orientation of the  $x, y, z$  axes (`rotate`). The `PUSH` command is then used to save the origin or axis orientation while the `POP` command is used to restore the origin and axis orientation.

**POP** The command, `pop`, restores the origin and axis orientation saved using a previous `pop` command. The total number of `pop` and `push` commands must be equal, otherwise a fatal error would occur. Smokeview detects this problem and draws a red sphere instead of the errantly defined object.

**PUSH** The command, `push`, saves the origin and axis orientation. (see above comment about number of `push` and `pop` commands).

**ROTATEX** The command, `r rotatex`, rotates objects drawn afterwards  $r$  degrees about the  $x$  axis.

**ROTATEY** The command, `r rotatey`, rotates objects drawn afterwards  $r$  degrees about the  $y$  axis.

**ROTATEZ** The command, `r rotatez`, rotates objects drawn afterwards  $r$  degrees about the  $z$  axis. A cone or any object for that matter may be drawn upside down by adding a `rotatez` command as in `180 rotatez 1.0 0.5 drawcone`.

**SCALEXYZ** The command, `x y z scalexyz`, stretches objects drawn afterwards by  $x, y$  and  $z$  respectively along the  $x, y$  and  $z$  axes. The `scalexyz` along with the `drawsphere` commands would be used to draw an ellipsoid by stretching a sphere along one of the axes.

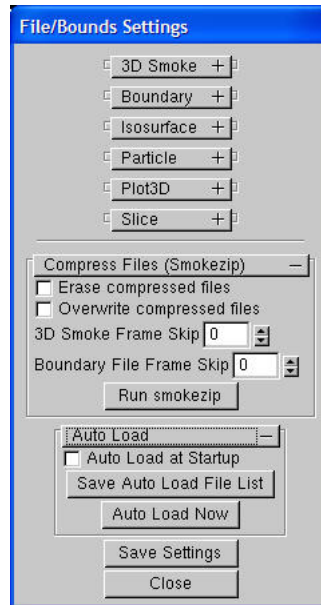


Figure 4.20: File/Bounds dialog box showing compression and autoload options. 3D smoke boundary files may be compressed using smokezip. All currently loaded files may be loaded automatically when smokeview first starts by selecting the autoload checkbox.

**SCALE** The command, `xyz scale`, stretches objects drawn afterwards `xyz` along each of the `x`, `y` and `z` axes (equivalent to `xyz xyz xyz scalexyz`).

**SETBW** The command, `grey setbw`, sets the red, green and blue components of color to grey (equivalent to `grey grey grey setcolor`). As with the `setcolor` command, `setbw` is only required when the grey level changes, not for each object drawn.

**SETCOLOR** The command, `r g b setcolor`, sets the red, green and blue components of the current color. Any object's drawn afterwards will be drawn with this color. That is, a `setcolor` command is not required for each object part drawn.

**SETLINEWIDTH** The command, `w setlinewidth` sets the width of lines drawn with the `drawline` and `drawcircle` commands.

**SETPOINTSIZE** The command, `s setpointsize`, sets the size of points drawn with the `drawpoint` command.

**TRANSLATE** The command, `x y z translate`, translates objects drawn afterwards by `x`, `y` and `z` along the `x`, `y` and `z` axes respectively. Equivalently, one can think of think of `x y z translate` as translating the origin by `(-x,-y,-z)`.

## 4.12 Smokezip - Compressing 3D Smoke and Boundary Files

3D smoke and boundary files may now be compressed using the stand-alone program smokezip or from within Smokeview using the compression menu item found on the Load/Unload menu. File compression may also be activated from the Compressions, Autoload section of the File Bounds dialog box illustrated in Figure 4.20. Compression is performed using the ZLIB compression library (see <http://www.zlib.org>). When using Smokeview, compression is performed in the *background* allowing one to continue visual-

ization. Smokeview adds the label *ZLIB* to Load menu entries for any files that have been compressed. Smokezip adds the extension svz to the corresponding compressed file name.

The usage for Smokezip is

```
smokezip 1.0 - Mar 12 2006
```

```
Compresses Smokeview 3D smoke and boundary files
```

```
smokezip [-f] [-s sourcedir] [-d destdir] [-c] casename [destdir]
```

- f** - overwrites compressed boundary and 3d smoke files
- d destdir** - if specified will copy compressed files (and files needed by Smokeview to view the case) to the directory destdir
- s sourcedir** - if specified assumes all files are in directory sourcedir
- c** - clean or remove all compressed files
- casename** - Smokeview .smv file

Smokezip uses min and max specifications found in the casename.ini file to determine how to map the four byte floating point data found in the boundary files to one byte color indices used by Smokeview. Smokezip uses percentile min and max bounds if the casename.ini file does not exist. The algorithms for determining the data mappings used by Smokeview and Smokezip are identical so should result in the same views.



## Chapter 5

### Summary

Often fire modeling is looked upon with skepticism because of the perception that eye-catching images shroud the underlying physics. However, if the visualization is done well, it can be used to assess the quality of the simulation technique. The user of FDS chooses a numerical grid on which to discretize the governing equations. The more grid cells, the better but more time-consuming the simulation. The payoff for investing in faster computers and running bigger calculations is the proportional gain in realism manifested by the images. There is no better way to demonstrate the quality of the calculation than by showing the realistic behavior of the fire.

Up to now, most visualization techniques have provided useful ways of analyzing the output of a calculation, like contour and streamline plots, without much concern for realism. A rainbow-colored contour map slicing down through the middle of a room is fine for researchers, but for those who are only accustomed to looking at real smoke-filled rooms, it may not have as much meaning. Good visualization needs to provide as much information as the rainbow contour map but in a way that speaks to modelers and non-modelers alike. A good example is smoke visibility. Unlike temperature or species concentration, smoke visibility is not a local quantity but rather depends on the viewpoint of the eye and the depth of field. Advanced simulators and games create the illusion of smoke or fog in ways that are not unlike the techniques employed by fire models to handle thermal radiation. The visualization of smoke and fire by Smokeview is an example of the graphics hardware and software actually computing results rather than just drawing pretty pictures. A common concern in the design of smoke control systems is whether or not building occupants will be able to see exit signs at various stages of a fire. FDS can predict the amount of soot is located at any given point, but that doesn't answer the question. The harder task is to compute on the fly within the visualization program what the occupant would see and not see. In this sense, Smokeview is not merely a *post-processor*, but rather an integral part of the analysis.



# Bibliography

- [1] K.B. McGrattan and G.P. Forney. Fire Dynamics Simulator (Version 5), User's Guide. NIST Special Publication xxxx, National Institute of Standards and Technology, Gaithersburg, Maryland, July 2007.
- [2] K.B. McGrattan (editor). Fire Dynamics Simulator (Version 5), Technical Reference Guide. NIST Special Publication xxxx, National Institute of Standards and Technology, Gaithersburg, Maryland, July 2007.
- [3] G.P. Forney and K.B. McGrattan. User's Guide for Smokeview Version 4 - A Tool for Visualizing Fire Dynamics Simulation Data. NIST Special Publication 1017, National Institute of Standards and Technology, Gaithersburg, Maryland, August 2004.
- [4] D. Madrzykowski and R.L. Vettori. Simulation of the Dynamics of the Fire at 3146 Cherry Road NE, Washington, DC May 30, 1999. Technical Report NISTIR 6510, Gaithersburg, Maryland, April 2000. URL: <http://fire.nist.gov/6510>.
- [5] D. Madrzykowski, G.P. Forney, and W.D. Walton. Simulation of the Dynamics of a Fire in a Two-Story Duplex, Iowa, December 22, 1999. Technical Report NISTIR 6854, Gaithersburg, Maryland, January 2002. URL: <http://www.fire.nist.gov/bfrlpubs/duplex/duplex.htm>.
- [6] R.L. Vettori, D. Madrzykowski, and W.D. Walton. Simulation of the Dynamics of a Fire in a One-Story Restaurant – Texas, February 14, 2000. Technical Report NISTIR 6923, Gaithersburg, Maryland, October 2002.
- [7] R.G. Rehm, W.M. Pitts, Baum H.R., Evans D.D., K. Prasad, K.B. McGrattan, and G.P. Forney. Initial Model for Fires in the World Trade Center Towers. Technical Report NISTIR 6879, Gaithersburg, Maryland, May 2002.
- [8] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. *OpenGL Programming Guide*. Addison-Wesley, Reading, Massachusetts, 3 edition, 1999.
- [9] Mark J. Kilgard. *OpenGL Programming for the X Window System*. Addison-Wesley Developers Press, Reading, Massachusetts, 1996.
- [10] Thomas Boutell. *CGI Programming in C & Perl*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1996.
- [11] Thomas Boutell. GD version 2.0.7, <http://www.boutell.com/gd/>, November 2002.
- [12] Guy Eric Schalnat, Andreas Dilger, and Glenn Randers-Pehrson. libpng version 1.2.5, <http://www.libpng.org/pub/png/>, November 2002.
- [13] jpeg version 6b, <http://www.ijg.org/>.

- [14] Jean loup Gailly and Mark Adler. zlib version 1.1.4, <http://www.gzip.org/zlib/>, November 2002.
- [15] Paul Rademacher. GLUI version 2.1, <http://www.cs.unc.edu/~rademach/glui/>.
- [16] Tomas Akenine-Moller and Eric Haines. *Real-Time Rendering*. A K Peters, Ltd., Natick, Massachusetts, 2nd edition, 2002.
- [17] K.B. McGrattan and G.P. Forney. Fire Dynamics Simulator (Version 4), User's Guide. NIST Special Publication 1019, National Institute of Standards and Technology, Gaithersburg, Maryland, July 2004.
- [18] Pamela P. Walatka and Pieter G. Buning. PLOT3D User's Manual, version 3.5. NASA Technical Memorandum 101067, NASA, 1989.
- [19] W. W. Jones, R. D. Peacock, G. P. Forney, and P. A. Reneke. CFAST, Consolidated Model of Fire Growth and Smoke Transport (Version 5. technical reference guide. NIST Special Publication 1030, National Institute of Standards and Technology, Gaithersburg, Maryland, October 2004.



## Appendix A

# Command Line Options

Smokeview may be run from a command shell. Furthermore, command line options may be invoked in order to alter Smokeview's startup behavior. Normally these options are not necessary. However, they may be used for cases with very large particle files or to generate a preference or customization file. To obtain a list of command line options, type:

```
smokeview -help
```

without any arguments which results in output similar to:

```
Smokeview 5.0.0 Beta - May 6 2007
Visualize fire/smoke flow simulations. All parameters are optional.
```

Usage:

```
smokeview casename -points m -frames n -ini -part -nopart -stereo -build
```

where

```
casename = project id (file names without the extension)
  m = maximum number of particles. Default=50000000
  n = maximum number of particle frames. Default=5001
  -ini = output default smokeview parameters to smokeview.ini
  -part = load particle file if present
  -nopart = do not load particle file
  -stereo = activate stereo mode (if supported)
  -build = show pre-preprocessing directives used to build smokeview
```

The `-nopart` option is used to prevent a particle file from being loaded at startup. The `-points` and `-frames` options are used to load more than the default 5,000,000 points or 500 frames where a frame is data for one time step. To load up to 6,000,000 points and 1000 frames then type:

```
smokeview casename -points 6000000 -frames 1000
```

where in both cases `casename` is the basename of the FDS output files. The same effect may be achieved by using:

```
MAXPOINTS
6000000
MAXFRAMES
1000
```

in the `smokeview.ini` or `casename.ini` file. This file may be created with the `-ini` option and contains many other customizable Smokeview parameters. The `-benchmark` option is used to measure the performance of Smokeview. The `-benchmark` option causes Smokeview to produce precise timings by outputting frame rates based upon one cycle through the timing loop rather than using moving averages. The `-stereo` option may be used to access stereo hardware (often called quad buffering) if it is available.

## Appendix B

# Menu Options

The design philosophy used to develop Smokeview has been to avoid complicated, non-portable user interfaces that are costly to implement. As a result, most of the development effort has gone into the visualization (display of particle flow, contour plots *etc*) rather than user interface design and implementation. Smokeview's pop-up menus are implemented with GLUT[9], the graphics library utility toolkit. The user interacts with Smokeview via 1) menus, 2) keyboard shortcuts and 3) the preference file (`smokeview.ini` or `casename.ini`).

A *pop up* menu is displayed when the right mouse is clicked anywhere within the scene. The main menu options as illustrated in Figure B.1 are: `Load/Unload`, `Show/Hide`, `Options`, `Dialogs`, `Tours`, `View`, `Help` and `Quit`. Several of these menu options have sub-menus. These menus are described in the following sections.

### B.1 Main Menu Items

**Load/Unload** This menu option allows one to load or unload particle, isosurface, vector slice, slice, Plot3D or boundary files. These time dependent files may be viewed simultaneously, but not concurrently with the time independent Plot3D files. However, one boundary or one Plot3D file per mesh may be viewed at a time. Multiple slices for the same variable may be viewed simultaneously. This menu may also be used to load and create preference files `.ini` files and to re-read the `.smv` file. For more details see Appendix B.2.

**Show/Hide** This menu item allows one to show or hide the loaded data files and various scene attributes such as time/color bars, internal blockages *etc*. As a file type is shown or hidden (or loaded and unloaded), the color and time bars are changed to reflect the currently visible data files. More details are given in Appendix B.3.

**Options** This menu allows one to specify various smokeview options such as specifying frame rates, dumping the screen to a PNG or JPEG file, changing font sizes, selecting dialog boxes *etc*.

**Dialogs** This menu allows one to open/close various dialog boxes for setting data bounds, controlling *look* of the 3D smoke, specifying clip planes *etc*.

**Tour** This menu displays a list of tours stored in the `casename.ini` preference file. The manual tour gives control back to the user. Tours may be defined using the Tour dialog box.

**View** Resets the simulation scene to an alternate view. The three choices are 1) exterior view, 2) interior view or a 3) user defined view. A viewpoint may be saved by using this menu or by



Figure B.1: Main Menu

using the `Viewpoint` sub-menu of the `Options` menu. If a time file is visible then two sub-menus occur allowing one to reset the view back to the original position or the time bar back to the initial time.

**Help** Displays a list and explanation of keyboard equivalent commands.

**Quit** Quit Smokeview.

## B.2 Load/Unload

Six types of files may be visualized with Smokeview. These files are loaded using the `LOAD/UNLOAD` menus as illustrated in Figure B.2. They are particle, vector slice, slice, boundary, isosurface and Plot3D files. Note that vector slice animations use two or more slice files to display the animated vector slices. The format of the data contained in these files is described in Appendix D.3. A sub-menu is present under `Load/Unload` for each file type generated for the simulation. Selecting one of the files appearing in the sub-menu causes it to be loaded and then displayed. The data may be unloaded or freed by selecting an `Unload` menu item appearing under the file list. Selecting `Unload All` as expected will unload all files. To hide a data file, select the `Show/Hide` menu option corresponding to the file type to be hidden.

The Smokeview `.smv` file contains information about all data files appearing in the `Load/Unload` menu item. The FDS field modelling software creates this file. (See Appendix D.2 for documentation on the format of this file).

The character “\*” occurring before a file name in one of the sub-menus indicates that the file has already been loaded. If the file below is loaded but not visible, then use the appropriate `Show/Hide` option to make it visible.

**Open Smokeview (.smv)** This menu item allows one (on the PC) to use an open file dialog box to open a Smokeview scene (file with `.smv` extension). This menu is only active when one starts up Smokeview without first specifying a file. (One cannot open a new Smokeview file after one has already been opened.)

**3D Smoke File (.s3d)** This menu item allows one to load realistic smoke, fire or water spray files.

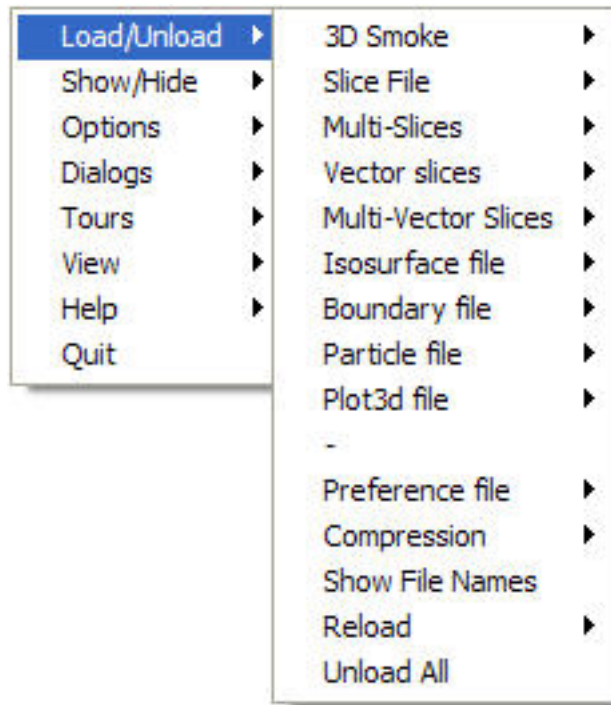


Figure B.2: Load/Unload Menu

**Slice File (.sf)** This menu item gives the name and location of all available slice files and also the option to unload the currently loaded slice files.

**Multi-Slice File (.sf)** This menu item allows one to load all slices occurring in one plane (within a grid cell) simultaneously. It also gives the option to unload the currently loaded multi-slices.

**Vector Slice File (.sf)** This menu item gives the name of all slice files that have one or more associated U, V and/or W velocity slice files. These slice files must be defined for the same region (or slice) in the simulation.

**Multi-Vector Slice File (.sf)** This menu item allows one to load all vector slices occurring in one plane (within a grid cell) simultaneously. It also gives the option to unload the currently loaded multi-slices.

**Isosurface File (.iso)** This menu item gives the name of all isosurface files and also the option to unload the currently loaded isosurface file.

**Boundary File (.bf)** This menu item gives the name of all boundary files and also the option to unload the currently loaded boundary file.

**Particle File (.part)** This menu item gives the name of all particle file and also the option to unload the currently loaded particle file.

**Plot3D File (.q)** This menu item gives the name of all Plot3D files and also the option to unload the currently loaded Plot3D file.

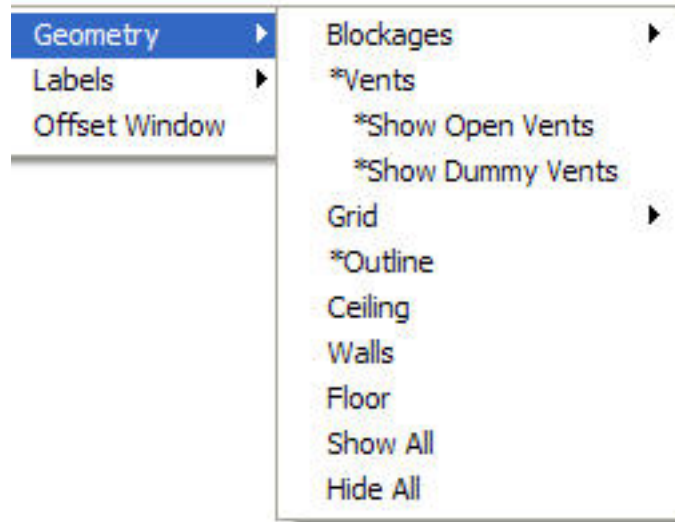


Figure B.3: Geometry Menu

**Preference File (.ini)** The INI or preference file contains configuration parameters that may be used to customize Smokeview's appearance and behavior. This menu item allows one to create (or overwrite) a preference file named either `smokeview.ini` or `casename.ini`. A preference file contains parameter settings for defining how Smokeview visualizes data. This file may be edited and re-read while Smokeview is running.

**Compression** 3D smoke and boundary files may be compressed using this menu item.

**Show File Names** Load and Unload menus by default are specified using the location and type of visual to be displayed. This menu item adds file names to the Load and Unload menus.

**Reload** This menu item allows one to reload files at intervals of 1, 5 or 10 minutes. This is useful when using Smokeview to display a case that is currently running in FDS.

**Unload All** This option causes all data files to be unloaded.

## B.3 Show/Hide

The `Show/Hide` menu item allows one to show or hide various parts of the simulation. This menu item contains sub-menus for Particles, Boundary, Animated Slice, Plot3D 2D and 3D contours, sensors (thermocouples, heat detectors and sprinklers), Color and time Bars and Geometry. These menu items only appear if they pertain to the simulation. For example the `Particles` sub-menu only appears if a particle file has been loaded. Similarly, the `Plot3D contouring` sub-menus only appear if a Plot3D file has been loaded. The "\*" character is used to indicate whether the visualization feature corresponding to the menu item is set or active.

### B.3.1 Geometry Options

The `Geometry` menu is illustrated in Figure B.3.

- Blockages** Blockage sub-menus are divided into two groups. The first group allows the user to change how blockage appear (`Defined in Input File`, `Normal`, `Outline`, `Hidden`). The second group allows the user to change where blockages are located (actual or requested). The `Actual` sub-menu positions blockages as computed by FDS (along grid lines). The `Requested` sub-menu positions blockages at locations as specified in the input file. If the `dx2dxf` conversion program was used to convert a Cad drawing to input compatible with FDS and Smokeview then a third menu option appears, `Cad`. This option displays the Smokeview scene in a form similar to the original Cad drawing.
- Grid** This option allows one to visualize the grid used to perform the numerical calculations. One selects `xy plane`, `xz plane` or `yz plane` to visualize a single plane or `Show All`, `Hide All` to show or hide all grids. (Keyboard shortcut: `g`)
- Outline** Show or hide the outline that frames the simulation scene.
- Floor** Show or hide the floor.
- Ceiling** Show or hide the ceiling.
- Vents** Show or hide the vents.
- Walls** Show or hide the walls.
- Show All** Show all geometric features described above.
- Hide All** Hide all geometric features described above.

### B.3.2 Animated Surface

This menu allows one to control the way isosurfaces are displayed. An isosurface is represented in Smokeview as a collection of triangles. Each triangle consists of three edges and three vertices.

- Solid** Display the isosurface by shading the triangles.
- Outline** Display the isosurface by only showing the triangle edges.
- Points** Display the isosurface by only showing the triangle vertices.
- quantity levels** Display the desired isosurface level (when more than one isosurface is stored in an isosurface file).
- Smooth** Display the isosurface using smoothed vertex normals.

### B.3.3 Particles

- Smoke (tracer)** Toggle the visibility of the particles. If water/sprinkler droplets are present in the particle file then sub-menus exist for both smoke and water to allow one to show or hide smoke and water droplets independently.
- Sprinkler** Toggle the visibility of sprinkler (water droplets).

### B.3.4 Boundary

- Exterior** Show or hide all data contained in a boundary (`.bf`) file pertaining to the exterior walls.

**Interior** Show or hide all data contained in a boundary (.bf) file pertaining to interior blockages.

**Front, Back, Left, Right, Up, Down** Toggle the visibility of whatever exterior boundary surface is selected. Note an exterior boundary menu option only appears if its data is present in the boundary file.

### B.3.5 Animated Vector Slice

Toggle the visibility of the animated vector slice file.

### B.3.6 Animated Slice

Toggle the visibility of the animated slice file.

### B.3.7 Plot3D

2D Contours

**Solution Variables** A Plot3D data file contains five solution variables. If one or more of the velocity components denoted  $u$ ,  $v$  and  $w$  are present in the Plot3D file then speed, calculated using  $\sqrt{u^2 + v^2 + w^2}$ , appears in the menu. Any velocity components missing from the Plot3D file are set to 0.0 when calculating speed. This menu item allows one to select the Plot3D solution variable to be visualized. (Keyboard shortcut: p)

**xy, xz, yz planes** These three menu items, appearing beneath the Solution Variables menu item, allow one to select which plane (xy, xz or yz) is displayed. (Applicable keyboard shortcuts: space bar, -, left/right cursor, up/down cursor, page up/down, 1 . . . 9)

**Flow vectors** Toggle visibility of flow vectors. The magnitude and direction of the vectors are determined by the U, V and W components of velocity. The vector color is determined by the solution variable selected. (Applicable keyboard shortcuts: a, s, v)

**Continuous** Display contours as smooth continuous shades or as stepped constant shades. (Keyboard shortcut: c)

**Show All** Show all three (xy, xz and yz) Plot3D planes at once.

**Hide All** Hide all three Plot3D planes.

Plot3D 3D Contours

**Solution Variables** Same as for 2D contours. This menu item allows one to select the solution variable used to generate the 3D or iso-contour to be displayed.

**Solution Value** Select the 3D contour level to display. The axis label shown in red corresponds to the 3D contour level displayed.

**Block Size** To increase the drawing speed, adjacent grid cells may be combined when viewing 3D contours. Selecting 1 will result in highly resolved contours but may take longer to draw. Selecting 5 will have the opposite effect.

**Hide** Hide the 3D contour.



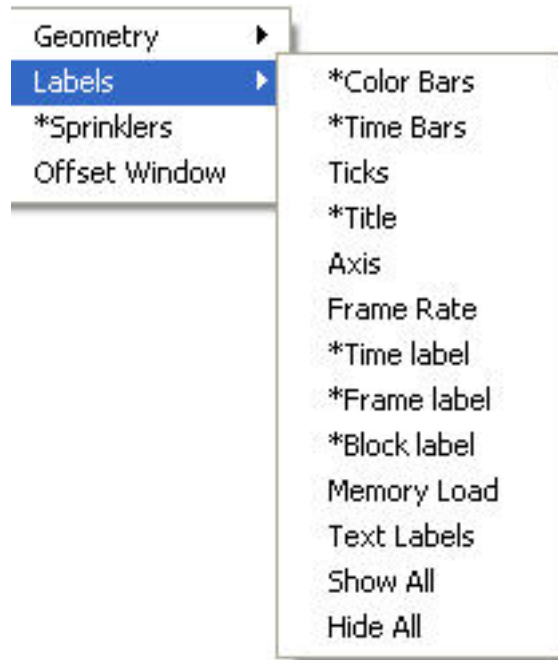


Figure B.4: Label Menu

### B.3.8 Heat detectors, Sprinklers, Thermocouples

Toggle the sensor visibility. The currently implemented sensors are heat detectors, sprinklers and thermocouples.

### B.3.9 Textures

Toggle the visibility of each or all textures.

### B.3.10 Labels

The label menu is illustrated in Figure B.4

**Color Bars, Time Bars, Title, Axis, Frame Rate, Time Label, Frame label, Mesh label, Memory Load, Text Labels**  
Show or hide the individual scene element.

**Show All** Show all scene elements.

**Hide All** Hide all scene elements.

## B.4 Options

The option menu is illustrated in Figure B.5 and detailed below.

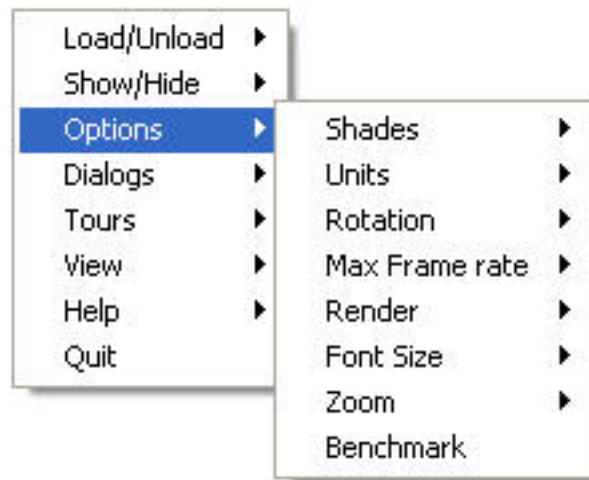


Figure B.5: Option Menu

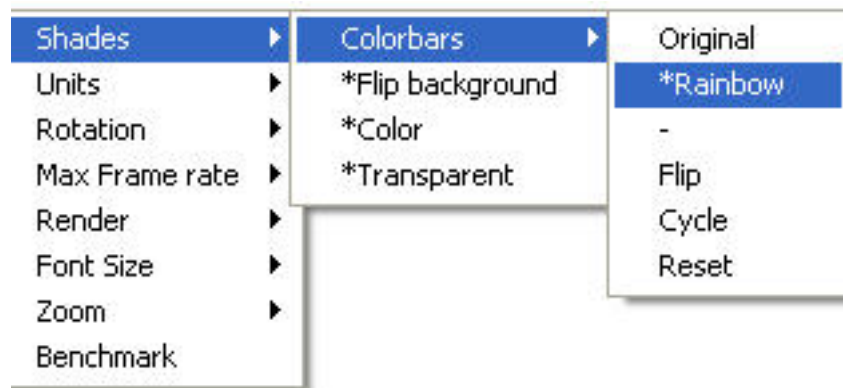


Figure B.6: Shades Menu

### B.4.1 Shades

The shades menu is illustrated in Figure B.6. This menu contains a new `colorbar` sub-menu which allows the user to choose and manipulate colorbars.

**Flip Background** Flip the background color between a dark and light shade. A dark background shade looks better on a computer monitor while a light shade looks better on the printed page.

**Flip Colorbar** Reverse the order of the colors displayed in the colorbar.

**Cycle Colorbar** Cycle the colors in the colorbar.

**Reset Colorbar** Return the colorbar to the original display.

**Color** Toggle scene colors between color and shades of gray. This option currently does not convert all scene colors to shades of gray. Contours and blockages are the most important Smokeview objects to convert.

**Transparent** Toggle contours between opaque and transparent. Transparent colors allow one to view the scene behind the contours giving the user a better sense of scale. Transparent colors, however, may make the scene look too confusing when the geometry is complex.

### B.4.2 Units

Select alternate temperature or velocity units. The `UNIT` keyword described in Appendix D.1 may be used to incorporate additional unit changes into Smokeview.

### B.4.3 Rotation

**Eye Centered** Rotate and move the scene relative to the observer's *eye*. Eye centered views make it easier to move around within the scene itself as in modern computer games. (Keyboard shortcut: `e`. The “e” keyboard shortcut toggles the view between an *eye centered* and a *world centered* perspective.)

**World Centered** Rotate and move the scene relative to the scene's center. (Keyboard shortcut: `e`. The “e” keyboard shortcut toggles the view between an *eye centered* and a *world centered* perspective.)

**World Centered - Level Rotation** As expected, this is the same as `World Centered` but with level rotations.

### B.4.4 Max Frame Rate

This option controls the rate at which image frames are displayed. The sub-menus allow one to specify a maximum frame rate. The actual frame rate may be slower if the scene is complex and the graphics card is unable to draw the scene sufficiently fast. The `unlimited` menu item allows one to display frames as rapidly as the graphics hardware permits. The `Real Time` menu item allows one to draw frames so that the simulation time matches real time. The `step` menu item allows one to step through the simulation one time step at a time. This menu item may be used in concert with the `Render` menu item described below to create images at the desired time and view orientation for inclusion into reports. This is how figures were generated in this report.

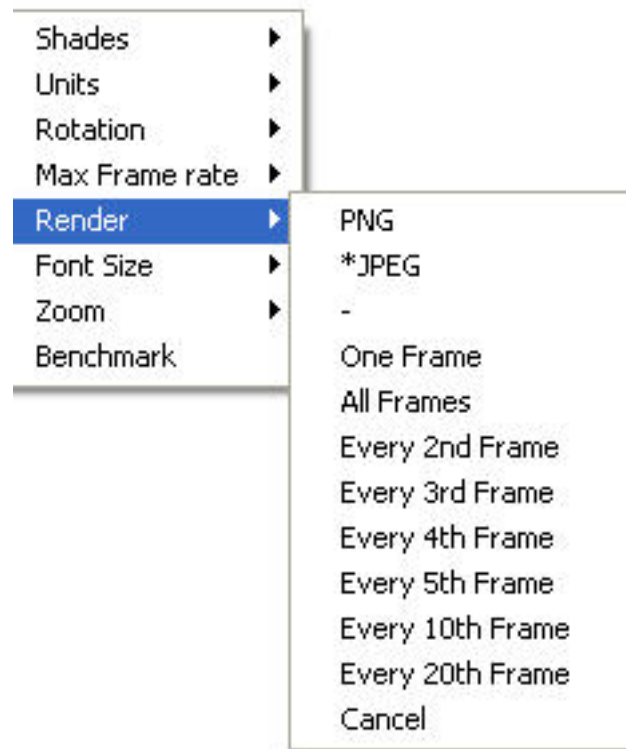


Figure B.7: Render Menu

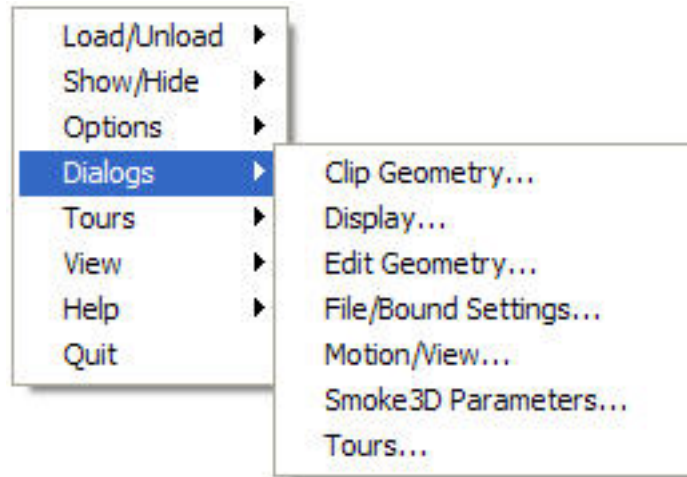


Figure B.8: Dialogs Menu

### B.4.5 Render

The **Render** menu, illustrated in Figure B.7, allows one to create PNG or JPEG image files of the currently displayed scene. The graphics library, GD 2.0.15, was used to create the rendered versions. GD 1.0 is documented in reference [10, Appendix 4]. GD 2.0.7 now creates images using *full color* allowing for more realistic scene representations eliminating the color banding that occurred with the previous version because of the limited number (256) of colors used to represent images. Due to patent disputes, GD 2.0.15 has dropped support for the GIF file format and uses JPEG or PNG instead.

The **Render** sub-menus allow one to specify an integer between 1 and 20 indicating the number of frames between rendered images. This allows one to generate images encompassing the entire time duration of the simulation which in turn can be converted into movie files (*mpeg, mov, avi etc*) using software available on the internet. Rendering may be stopped by selecting **Cancel**.

The keyboard shortcut for the render option is **r**.

### B.4.6 Font Size

This option allows one to display text in either a normal or a large font.

### B.4.7 Zoom

This menu item allows one to change the perspective by altering the magnification used to display the scene.

## B.5 Dialogs

The **Dialogs** menu, illustrated in Figure B.8, allows one to select dialog boxes for setting various Smoke-view features.

**Stereo 3-D** Provides stereo 3-D output on systems equipped with supported video cards. Stereo 3-D requires a computer graphics card, monitor, software drivers and shuttered glasses. This option

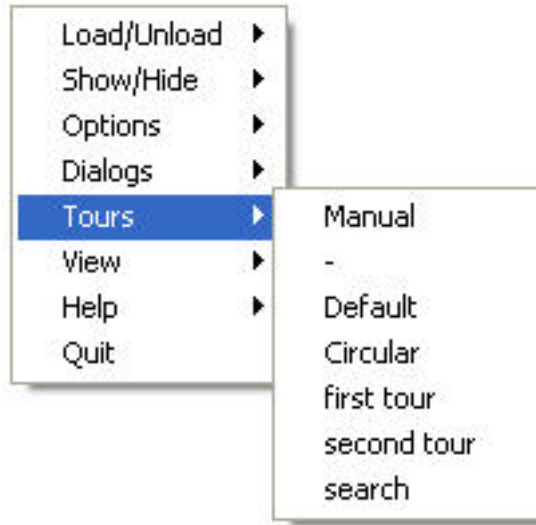


Figure B.9: Tour Menu

only appears if Smokeview is started from the command line using the `-stereo` command line option and if a video card supporting quad buffering (left and right front buffer, and left and right back buffer) is supported.

**Clip Geometry** Open the dialog box for clipping the geometry allowing one to see past exterior portions of the scene.

**Display** Open the dialog box for setting various parameters that control how the Smokeview scene appears.

**Edit Geometry** Open the dialog box for editing FDS blockages.

**File/Bound Settings** Open the dialog box for specifying data bounds.

**Motion/View** Open the dialog box for controlling scene movement. To use the movement arrows, click and hold the mouse in one of the arrows then move the mouse to achieve the desired movement.

**Smoke3D Parameters** Open the dialog box for specifying controlling the look of the 3D smoke.

**Tours** This menu item opens up the **Edit Tour** dialog box and shows the path over which the tours occur.

## B.6 Tours

The Tour menu illustrated in Figure B.9 allows one to show and hide available tours.

**Manual** This menu item turns *touring* off, allowing the user to control the scene with the mouse.

**Default** This menu item activates the original *tour* used in previous versions of Smokeview.

**List of tours** Each tour defined in the `.ini` file is listed under the Tour menu. A *circular* tour is defined automatically by Smokeview.

## Appendix C

# Keyboard Shortcuts

Many menu commands have equivalent keyboard shortcuts. These shortcuts are described here and are also briefly described under the Help menu item from within Smokeview itself.

- a** Alter the arrow size used to display velocity vectors.
- a** Slice left when in **eye centered** movement mode.
- b** To increase the drawing speed, adjacent grid cells may be combined into coarser pseudo-grid cells when generating and viewing 3D contours. The “b” key is used to increment the number of grid cells that are combined (between one and five grid cells) when generating 3D contours. Using one grid will result in highly resolved contours but may take longer to draw. Using five grid cells will have the opposite effect resulting in coarse contours that can be drawn quickly.
- c** Toggle the Plot3D 2D contour display between solid banded contours and continuously shaded contours.
- d** Slide right when in **eye centered** movement mode.
- e** Toggle how the scene is manipulated between an *eye view* where scene motion is relative to the observer and a *world view* where the scene motion is relative to the scene center.
- g** Toggle the grid visibility. When the grid display option is active, the x, y and z keys may be used to show or hide the grid perpendicular to the x, y and z axes respectively.
- h** Toggle window indentation, for use with window capturing.
- i** Toggle the visibility of iso-contours (3D surface contours).
- k** Toggle the visibility of the time bar.
- m** Switch between blocks in a case that has more than one mesh.
- o** Reset a time dependent animation to the initial time.
- p** Show the next variable in the Plot3D data set.
- P** The cursor keys and the page up/down keys are needed to both move grid and Plot3D planes around the scene and to move through the scene while in *eye centered* mode. The upper case “P” key is then used to switch the way these keys are used between grid/Plot3D movement mode and scene movement mode.
- q** Switch between blockage views. These views are blocks that are aligned on grid lines, blocks as specified by the user (in the .data file) and blocks as generated by a CAD (computer aided drawing) package.

- r,R** Render the current scene as a JPEG or a PNG file which can be viewed in a web browser or inserted into a word processing document. If the upper case R key is selected then the scene is rendered in with double the screen resolution.
- s** Increment the number of vectors skipped. This is useful for making vector displays more readable when grids are finely meshed.
- s** Move backwards when in **eye centered** movement mode.
- t** Toggle the time stepping mode. Time stepping mode allows one to step through the simulation one time step at a time.
- T** Toggle the texture method for drawing slice and boundary file colors.
- v** Toggle vector visibility. This option is only active when there are U, V and/or W velocity components present in the Plot3D data set.
- w** Move forwards when in **eye centered** movement mode.
- x, y, z** Toggle the visibility of the Plot3D data planes perpendicular to the x, y and z axes respectively (parallel to the yz, xz and xy planes).
- Left/Right Cursor** When the *eyevew* mode is **eye centered** then these keys rotate the scene to the left or right otherwise they increment/decrement the Plot3D plane location displayed in the xz plane.
- Page Up, Page Down** Increment/decrement the Plot3D plane location displayed in the xy plane.
- Up/Down Cursor** Increment/decrement the Plot3D plane location displayed in the yz plane.
- Decrement Plot3D data planes, Plot3D iso-contour levels or time step displayed.
- space bar** Increment Plot3D data planes, Plot3D iso-contour levels or time step displayed.



## Appendix D

# File Formats

### D.1 Smokeview Preference File Format (.ini files)

Smokeview uses preference files to set input parameters not settable by using menus or the keyboard and to save the *state* of a visualization. Smokeview looks for preference files in three locations in the following order:

1. a file named `smokeview.ini` in a global directory defined by the `SMOKEVIEWINI` environment variable. On the PC, the directory `C:\nist\fds\smokeview.ini` is the default location for this preference file. The `SMOKEVIEWINI` environment variable may be defined on the PC by adding the line:

```
set SMOKEVIEWINI=dir
```

to the `C:\autoexec.bat` file where `dir` is the directory where the global preference file is stored. This step is performed automatically by the Smokeview installation program. This environment variable may be defined on a UNIX workstation by adding the line:

```
setenv SMOKEVIEWINI dir
```

to a `.login` or `.cshrc` start up file again where `dir` is the directory containing the global preference file. Changes to this `smokeview.ini` file apply to all cases visualized on the computer unless overridden by preference files named or located in directories named in steps 2. and 3.

2. a file named `smokeview.ini` in the directory containing the case being visualized. Changes to this `smokeview.ini` file apply to all cases in the current directory unless overridden by the `casename.ini` file contained in this directory.
3. a file named `casename.ini` in the directory containing the case being visualized where `casename` is the name of the case.

The `smokeview.ini` file may be created by typing:

```
smokeview -ini
```

from the command line or by selecting the `Write smokeview.ini` menu item. The `casename.ini` preference file can be created via the menus or by copying a previously created `smokeview.ini` file.

Smokeview reads the global `smokeview.ini` file first (step 1. above), followed by the local `smokeview.ini` file (step 2. above), followed by the `casename.ini` file. The global `smokeview.ini` file is used to customize parameters for all Smokeview runs. The local `smokeview.ini` file is used to customize parameters for just those Smokeview runs contained in the local directory. The `casename.ini` file is used to customize parameters for only those Smokeview runs with the prefix `casename`.

All preference file parameters unless otherwise noted consist of a **KEYWORD** followed by a value, as in:

```
KEYWORD
value
```

### D.1.1 Color parameters

All colors are specified using a 3-tuple: `r g b` where `r`, `g` and `b` are the red, green and blue components of the color respectively. Each color component is a floating point number ranging from 0.0 to 1.0 where 0.0 is the darkest shade and 1.0 is the lightest shade. For example the 3-tuple `1.0 0.0 0.0` is bright red, `0.0 0.0 0.0` is black and `1.0 1.0 1.0` is white.

**AMBIENTLIGHT**      Sets the color used for specifying ambient light. (default: `0.6 0.6 0.6`)

**BACKGROUNDCOLOR**      Sets the color used to visualize the scene background. (default: `0.0 0.0 0.0`)

**BLOCKCOLOR**      Sets the color used to visualize internal blockages. (default: `1.0 0.8 4.0`)

**BOUNDCOLOR**      Sets the color used to visualize floors, walls and ceilings. (default: `0.5 0.5 0.2`)

**COLORBAR**      Entries for the color palette in RGB (red, green, blue) format where each color component ranges from 0.0 to 1.0 . The default values (rounded to 2 digits) are specified with:

```
COLORBAR
12
0.00 0.00 1.00
0.00 0.28 0.96
0.00 0.54 0.84
0.00 0.76 0.65
0.00 0.91 0.41
0.00 0.99 0.14
0.14 0.99 0.00
0.41 0.91 0.00
0.65 0.76 0.00
0.84 0.54 0.00
0.96 0.28 0.00
1.00 0.00 0.00
```

**COLOR2BAR**      Miscellaneous colors used by Smokeview. The default values are specified using:

```
COLOR2BAR
8
1.0 1.0 1.0 :white
```

```

1.0 1.0 0.0 :yellow
0.0 0.0 1.0 :blue
1.0 0.0 0.0 :red
0.0 1.0 0.0 :green
1.0 0.0 1.0 :magenta
0.0 1.0 1.0 :cyan
0.0 0.0 0.0 :black

```

where the 8 indicates the number of colors defined and the character string after the ``:'' are ignored.

**COLORBARFLIP** Specifies whether the colorbar is flipped (1) or not flipped (0) (default: 0).

**DIFFUSELIGHT** Sets the color for specifying diffuse light (default: 0.5 0.5 0.5).

**FLIP** Specifies whether to flip (1) or not to flip (0) the foreground and background colors. By default the background color is black and the foreground color is white. Setting FLIP to 1 has the effect of having a white background and black foreground. (default: 0).

**FOREGROUNDCOLOR** Sets the color used to visualize the scene foreground (such as text labels). (default: 1.0 1.0 1.0)

**HEATOFFCOLOR** Sets the color used to visualize heat detectors before they activate. (default: 1.0 0.0 0.0)

**HEATONCOLOR** Sets the color used to visualize heat detectors after they activate. (default: 0.0 1.0 0.0)

**ISOCOLORS** Colors and parameters used to display animated isocontours. Default:

```

ISOCOLORS
10.000000 0.800000 : shininess, transparency
0.700000 0.700000 0.700000 : specular
3
0.960000 0.280000 0.000000
0.750000 0.800000 0.800000
0.000000 0.960000 0.280000

```

**SENSORCOLOR** Sets the color used to visualize sensors. (default: 1.0 1.0 0.0)

**SETBW** The parameter used to set whether color shades (0) or shades of grey (1) are to be used for coloring contours and blockages. (default: 0)

**SPRINKOFFCOLOR** Sets the color used to visualize sprinklers before they activate. (default: 1.0 0.0 0.0)

**SPRINKONCOLOR** Sets the color used to visualize sprinklers after they activate. (default: 0.0 1.0 0.0)

**STATICPARTCOLOR** Sets the color used to visualize static particles (particles displayed in frame 0). (default: 0.0 1.0 0.0).

**TIMEBARCOLOR** Sets the color used to visualize the timebar. (default: 0.6 0.6 0.6)

**VENTCOLOR** Sets the color used to visualize vents. (default: 1.0 0.0 1.0)

## D.1.2 Size parameters

The parameters described in this section allow one to customize the size of various Smokeview scene elements.

**LINEWIDTH** Defines the width of lines<sup>1</sup> in pixels. (default: 2.0)

**PARTPOINTSIZ** Defines the size in pixels of smoke or tracer particles. (default: 1.0)

**SENSORABSSIZE** Defines the sensor size drawn by smokeview using the same units as used to specify the grid coordinates. (default: 0.038)

**SLICEOFFSET** Defines an offset distance<sup>2</sup> animated slices are drawn from adjacent solid surfaces. (default: 0.10)

**SMOOTHLINES** Specifies whether lines should be drawn (1) or not drawn (0) using anti-aliasing (default: 1).

**SPRINKLERABSSIZE** Defines the sprinkler size drawn by smokeview using the same units as used to specify the grid coordinates. (default: 0.076)

**STREAKLINEWIDTH** Defines the width of a streak line. (default: 1.0)

**VECTORLENGTH** Defines the length of Plot3D vectors. A vector length of 1.0 fills one grid cell. Vector lengths may also be changed from within Smokeview by depressing the “a” key. (default: 4.0)

**VECTORPOINTSIZ** Defines the size in pixels of the point that is drawn at the end of a Plot3D vector. (default: 3.0)

**VENTLINEWIDTH** Defines the width of lines used to draw vents in pixels. (default: 2.0)

**VENTOFFSET** Defines a distance used to offset vents drawn from adjacent surfaces. (default: 0.10 (units of fraction of a grid cell width))

**WINDOWHEIGHT** Defines the initial window height in pixels. (default: 480)

**WINDOWWIDTH** Defines the initial window width in pixels. (default: 640)

**WINDOWOFFSET** Defines a margin offset around the Smokeview scene for use when capturing images to video. (default: 45)

## D.1.3 Time, Chop and value bound parameters

This section describes parameters used by Smokeview to 1) modify the time intervals used to load data (keywords beginning with T\_ ), 2) eliminate or chop data from being displayed (beginning with C\_) and 3) override the minimum and maximum data values (keywords beginning with V\_) used to convert data to color values. By default, Smokeview reads in data for all time values and does not override minimum and maximum data values. Each time and data bound keyword (except for V\_PLOT3D) has the format:

KEYWORD  
minflag minvalue maxflag maxvalue

---

<sup>1</sup>Except lines used to draw vents

<sup>2</sup>distance is relative to the maximum grid cell width

where `minflag` can be either 0 or 1. If it is 1 then, the subsequent number, `minvalue` is used by Smokeview to scale the data otherwise if `minflag` is 0 then `minvalue` is ignored. The next two parameters `maxflag` and `maxvalue` are defined similarly. The `V_PLOT3D` keyword contains data bound entries for each variable in the Plot3D data file. If a Plot3D *speed* variable was constructed by Smokeview then the `V_PLOT3D` keyword will contain six entries instead of five.

**C\_PARTICLES** Defines the minimum and maximum values used to discard particle data in a visualization. To drop particle data below 70°C and above 200°C use:

```
C_PARTICLES 1 70. 1 200.
```

**C\_PLOT3D** Defines the minimum and maximum data values used to drop or chop Plot3D. To cause Smokeview to set the minimum and maximum chop values for the first PLOT3D quantity (usually temperature) to 100 and 300 use:

```
C_PLOT3D 5
1 1 100.0 1 300.0
2 0 1.0 0 0.0
3 0 1.0 0 0.0
4 0 1.0 0 0.0
5 0 1.0 0 0.0
```

The integer “1” occurring before the “100” or “300” causes Smokeview to use the next number as a minimum or maximum chop value respectively.

**C\_SLICE** Defines the minimum and maximum values used to discard slice file data in a visualization. To drop slice data below 70°C and above 200°C use:

```
C_SLICE 1 70. 1 200.
```

**T\_BOUNDARY** Defines the minimum and maximum times used to include boundary frames in a visualization. To load boundary data between 20 and 150 seconds use:

```
T_BOUNDARY
1 20. 1 150.
```

(default: 0 1.0 0 0.0)

**T\_ISO** Defines the minimum and maximum times used to include isosurface frames in a visualization. To load isosurface data between 20 and 150 seconds use:

```
T_ISO
```

```
1 20. 1 150.
```

**T\_PARTICLES** Defines the minimum and maximum times used to include particles in a visualization. To load particle data between 20 and 150 seconds use:

```
T_PARTICLES
1 20. 1 150.
```

**T\_SLICE** Defines the minimum and maximum times used to include slice frames in a visualization. To load slice data between 20 and 150 seconds use:

```
T_SLICE
1 20. 1 150.
```

(default: 0 1.0 0 0.0)

**V\_BOUNDARY** Defines the minimum and maximum data bounds used to convert boundary data values to color indices. (default: 0 1.0 0 0.0)

The **V\_BOUNDARY** keyword has an optional parameter allowing one to specify to which type of data the bounds should apply. For example, to specify boundary file bounds for temperature (30.0 °C, 600.0 °C) use:

```
V_BOUNDARY
1 30.000000 1 600.000000 TEMP
```

where **TEMP** is the Smokeview colorbar labels displayed when showing the boundary file.

These suffixes are added automatically when the Bounds dialog box is used to set data bounds.

**V\_PARTICLES** Defines the minimum and maximum data bounds used to convert particle data values to color indices. (default: 0 1.0 0 0.0)

**V\_PLOT3D** Defines the minimum and maximum data bounds used to convert Plot3D data values to color indices. The default values are given by:

```
V_PLOT3D
5
1 0 1.0 0 0.0
2 0 1.0 0 0.0
3 0 1.0 0 0.0
4 0 1.0 0 0.0
5 0 1.0 0 0.0
```

where the initial 5 indicates the number of subsequent entries, the first integer on each line

indicates the Plot3D data variable being specified and all other parameters on each line are defined as above.

To cause Smokeview to set the minimum and maximum data values to for the first quantity (usually temperature) to 20 and 600 use:

```
V_PLOT3D
5
1 1 20.0 1 600.0
2 0 1.0 0 0.0
3 0 1.0 0 0.0
4 0 1.0 0 0.0
5 0 1.0 0 0.0
```

The integer “1” occurring before the “20” or “600” causes Smokeview to use the next number as a minimum or maximum value respectively otherwise if “0” is specified then Smokeview ignores the subsequent min/max value.

In addition to 0 and 1, the V\_PLOT3D keyword may use 2 as a bound indicator. In the above example, if 2 rather than 1 is used to define Plot3D bounds, then Smokeview does not draw contour<sup>3</sup> levels smaller than 20 or contours greater than 600. The PLOT3D bound line 1 2 360.0 1 420.0 indicates that temperatures below 360 °C are not drawn and that temperatures above 420 °C are drawn with the *highest* color (black in black and white mode, red in color mode). This bound line could also be implemented with the *Set Bounds* dialog box as illustrated in Figure ?? resulting in a contour plot as illustrated in Figure 4.8.

**V\_SLICE** Defines the minimum and maximum data bounds used to convert slice data values to color indices. (default: 0 1.0 0 0.0)

The V\_SLICE keyword has an optional parameter allowing one to specify to which type of data the bounds should apply. For example, to specify separate slice file bounds for temperature (30.0 °C, 600.0 °C) and the U component of velocity (-1.0 m/s and 1.0 m/s) use:

```
V_SLICE
1 30.000000 1 600.000000 TEMP
V_SLICE
1 -1.0 1 1.0 U-VEL
```

where TEMP and U-VEL are the Smokeview colorbar labels displayed when showing the slice file.

These suffixes are added automatically when the Bounds dialog box is used to set data bounds.

#### D.1.4 Data loading parameters

The keywords in this section may be used to reduce the memory required to visualize FDS data. Keywords exist for limiting particles and frames. Other keywords exist for compressing particle data and skipping

---

<sup>3</sup>This is true for *stepped* or discrete contours. If *continuous* contours are drawn, then “2” and “1” have the same effect.

particles and frames.

**BOUNDFRAMESTEP** Defines the number of intervals or steps between loaded boundary file frames. (default: 1)

**EVACFRAMESTEP** Specifies the interval or steps between loaded evacuation frames. An evacuation frame consists of all the *people* loaded at one particular time step. For example, if EVACFRAMESTEP is set to 3 then every 3rd evacuation frame is read in. (default: 1)

**ISOFRAMESTEP** Defines the number of intervals or steps between loaded isosurface file frames (default: 1)

**MAXFRAMES** Specifies the maximum number of particle frames that can be read in. (default: 501)

**MAXPOINTS** Specifies the maximum number of particle points that can be read in from a particle file. (default: 5000000)

**NOPART** Indicates that a particle file should not (1) or should (0) be loaded when Smokeview starts up. This option is used when one wants to look at other files besides the particle file. (default: 0)

**PARTFRAMESTEP** Specifies the interval or steps between loaded particle frames. A particle frame is all the particles loaded at one particular time step. For example, if PARTFRAMESTEP is set to 3 then every 3rd particle frame is read in. This and the PARTPOINTSTEP options may be used together to dramatically reduce the memory required to view particle files. These options should be used when displaying Smokeview files on computers with limited memory. (default: 1)

**PARTPOINTCOMPRESS** Specifies how Smokeview stores particle data. Each particle is represented using three position coordinates and one data value. Using full precision (PARTPOINTCOMPRESS value of 0), Smokeview uses four bytes per position coordinate and one byte for the data value or 13 bytes to represent each particle point. If this keyword is set to 1 then Smokeview uses one byte per position coordinate and one byte for the data value or 4 bytes per particle point. Finally if this keyword has value 2 then 2 bytes are used to store each position coordinate and one byte for the data value or 7 bytes per particle point. The recommended value is 2. Using 0 results in almost twice the memory usage without any extra precision in positioning particles. Using 1, though using less memory than the “2” option, results in granularity effects when displaying particles which the user may find annoying. (default: 2)

**PARTPOINTSTEP** Specifies the interval or steps between loaded particle points. For example, if PARTPOINTSTEP is set to 5 then every 5th particle or only 20 per cent of the particle file is read in. (default: 1)

**SLICEFRAMESTEP** Specifies the number of interval or steps between loaded slice file frames. (default: 1)

**SMOKE3DFRAMESTEP** Specifies the number of interval or steps between loaded 3D smoke file frames. (default: 1)

**SMOKE3DZIPSTEPxxx**

**SLICEDATAOUT** When set to 1 will output data corresponding to any loaded slice files whenever the scene is rendered.



### D.1.5 Viewing parameters

The keywords in this section define how a scene is viewed. Keywords exist for showing or hiding various scene elements and for modifying how various scene elements appear.

**APERTURE** Specifies the viewing angle used to display a Smokeview scene. Viewing angles of 30, 45, 60, 75 and 90 degrees are displayed when APERTURE has the value of 0, 1, 2, 3 and 4 respectively. (default: 2)

**AXISNUM** Specifies whether Smokeview should (AXISNUM set to 1) or should not (AXISNUM set to 0) calculate data minimums and maximums based upon statistical limits. For example, if this parameter is set to 1 then Smokeview will set the minimum and maximum data bound to be the 1st and 99th percentile of the data respectively. (default: 1)

**AXISSMOOTH** Specifies whether axis numerical labels should be smoothed (AXISSMOOTH set to 1) or not smoothed (AXISSMOOTH set to 0). (default: 1)

**BLOCKLOCATION** Specifies the location or method used to draw blockages. Blockages are drawn either snapped to the nearest grid (5), drawn at locations as specified in the FDS input file (6) or drawn as specified in a compatible CAD package (7). A program used to convert DXF formatted CAD files to files compatible with FDS and Smokeview is available at <http://fire.nist.gov/fds/ref>. (default: 5)

**CLIP** Specifies the near and far clipping plane locations. Dimensions are relative to the longest side of an FDS scene. (default: 0.001 3.000)

**CULLFACES** Hide (1) or show (0) the back side of various surfaces.

**EYEVIEW** Specifies whether the scene should be rotated relative to the observer (EYEVIEW set to 1) or the scene center (EYEVIEW=0). (default: 0)

**EYEX, EYEV, EYEZ** The parameters EYEX, EYEV, EYEZ specify the x, y and z location respectively of the viewing origin (where your eyes are). (default: 0.5 -0.9 1.5)

**FONTSIZE** Specifies whether small (0) or large (1) fonts should be used to display text labels. (default: 0)

**FRAMERATEVALUE** Specifies the maximum rate (frames per second) that Smokeview should display frames. This value is an upper bound. Smokeview will not display frames faster than this rate but may display frames at a slower rate if the scene to be visualized is complex. (default: 1000000 (essentially unlimited))

**MSCALE** Specifies how dimensions along the X, Y and/or Z axes should be scaled. (default: 1.0 1.0 1.0)

**PROJECTION** Specifies whether a perspective (0) or orthographic (1) projection is used to draw Smokeview scenes. (default: 0)

**P3CONT2D** The parameter P3CONT2D may be set to 0, 1 or 2. If P3CONT2D is set to 0 then Plot3D color contours are drawn by coloring each node and letting OpenGL interpolate colors between nodes. If P3CONT2D is set to 1 then discrete or stepped shaded contours are drawn. If P3CONT2D is set to 2 then contour lines are drawn. (default: 1)

**P3DSURFACETYPE** Specifies how Plot3D isosurfaces should be drawn. If P3DSURFACETYPE is set to 1 then Plot3D isosurfaces are drawn using shaded triangles. If P3DSURFACETYPE is set to 2 or 3 then Plot3D isosurfaces are drawn using triangle outlines and points respectively. (default: 1)

**P3DSURFACESMOOTH** When drawing Plot3D isosurfaces using shaded triangles, this option specifies whether the vertex normals should be averaged (P3DSURFACESMOOTH set to 1) resulting in smooth isosurfaces or not averaged resulting in isosurfaces that have sharp edges (P3DSURFACESMOOTH set to 0). (default: 1)

**RENDERFILETYPE** Specifies whether PNG (RENDERFILETYPE set to 0) or JPEG (RENDERFILETYPE set to 1) should be used to render images. (default: 1)

**RENDEROPTIONxxx**

**SHOWAXISLABELS** Specifies whether axis labels should be drawn (1) or not drawn (0) drawn. (default: 0)

**SHOWBLOCKLABEL** Specifies whether a label identifying the active mesh should be drawn (1) or not drawn (0). (default: 1)

**SHOWBLOCKS** Specifies how a blockage should be drawn. A value of 0, 1 or 2 indicates that the blockages are invisible, drawn normally or drawn as outlines respectively. (default: 1)

**SHOWCADANDGRID**

**SHOWCEILING** Specifies whether the ceiling (upper bounding surface) should be drawn (1) or not drawn (0). (default: 0)

**SHOWCOLORBARS** Specifies whether the color bars should be drawn (1) or not drawn (0). (default: 1)

**SHOWFLOOR** Specifies whether the floor (lower bounding surface) should be drawn (1) or not drawn (0). (default: 1)

**SHOWFRAME** Specifies whether the frame surrounding the scene should be drawn (1) or not drawn (0). (default: 1)

**SHOWFRAMELABEL** Specifies whether the frame number should be drawn (1) or not drawn (0). (default: 1)

**SHOWFRAMERATE** Specifies whether the frame rate label should be drawn (1) or not drawn (0). (default: 0)

**SHOWGRIDLOC** Specifies where grid location should be drawn (1) or not drawn (0). (default: 0)

**SHOWIGNITION** When drawing a temperature boundary file, this option specifies whether ignited materials (regions exceeding the materials ignition temperature) should be drawn (1) or not drawn (0). A second parameter specifies whether only the ignited regions should be drawn (1) or both the ignited regions and other regions should be drawn (0). (default: 0 0)

**SHOWLABELS** Specifies whether labels should be drawn (1) or not drawn (0). Labels are specified using the **LABEL** keyword described in subsection 4.10. (default: 0)

**SHOWMEMLOAD** Specifies (when run on a PC) whether a label giving the memory used should be drawn (1) or not drawn (0). (default: 0)

**SHOWNORMALWHENSMOOTH** Specifies that smooth blocks should be drawn as normal blocks (1) or drawn as smooth blocks (0). (default: 0)

**SHOWOPENVENTS** Specifies that open vents should be drawn (1) or not drawn (0). (default: 0)

**SHOWDUMMYVENTS** Specifies that dummy vents (vents created by FDS) should be drawn (1) or not drawn (0). (default: 0)

**SHOWSENSORS** Specifies whether sensors should be drawn (1) or not drawn (0). A second parameter specifies whether the sensor's orientation or normal vector should be drawn (1) or not drawn (0). (default: 1 0).

**SHOWSLICEINOBST** Specifies whether a slice file should be drawn (1) inside a blockage or not drawn (0) inside a blockage. Normally a slice file is not drawn inside a blockage but one would want to draw a slice file inside a blockage if the blockage disappears over the duration of a run. (default: 0).

**SHOWSMOKEPART** Specifies whether smoke or trace particles should be drawn (1) or not drawn (0). (default: 1)

**SHOWSPRINKPART** Specifies whether sprinkler droplet particles (if present in the particle file) should be drawn (1) or not drawn (0). (default: 1)

**SHOWTICKS** Specifies whether labels should be drawn (1) or not drawn (0). Ticks are specified using the **TICK** keyword described in subsection 4.10. (default: 0)

**SHOWTIMEBAR** Specifies whether the timebar should be drawn (1) or not drawn (0). (default: 1)

**SHOWTIMELABEL** Specifies whether the time label should be drawn (1) or not drawn (0). (default: 1)

**SHOWHMSTIMELABEL** Specifies whether the time label should be drawn (1) or not drawn (0) using the format "h:m:s" where "h" is hours, "m" is minutes and "s" is seconds. (default: 0)

**SHOWTITLE** Specifies whether the title should be drawn (1) or not drawn (0). (default: 1)

**SHOWVENTS** Specifies whether vents should be drawn (1) or not drawn (0). (default: 1)

**SHOWWALLTEXTURES** If wall textures are defined in the input .smv file, this option specifies whether to draw (1) or not to draw (0) wall textures. (default: 0)

**SHOWWALLS** Specifies whether the four walls (four vertical bounding surfaces) should be drawn (1) or not drawn (0). (default: 1)

**STARTUPVIEW** Used by Smokeview to record the view to be displayed at startup (as defined in a previous Smokeview run).

**SURFINC** Smokeview allows one to display two Plot3D isosurfaces simultaneously. The SURFINC parameter specifies the interval between displayed Plot3D surface values. (default: 0)

**TIMEOFFSET** Specifies that offset time in seconds added to the displayed simulation time. Along with the **SHOWHMSTIMELABEL** keyword, the **TIMEOFFSET** keyword allows one to display *wall clock* rather than simulation time. (default: 0.0)

**TITLESAFE** Amount in pixels to offset titles when displaying scene in *title safe* mode. (default: 0)

**TRANSPARENT** Specifies whether 2D and 3D contours should be drawn with solid colors (0) or transparent colors(1). (default: 1)

**TWOSIDEDVENTS** Specifies whether to draw vents so that they are visible from both sides (1) or visible from only one side (0). (default: 0)

**UNIT** This keyword defines how to calculate colorbar label values for non-default units. Note that in the following example all text to the right of the '/' characters is not part of the keyword but is used to explain the format.  
default:

UNIT

```

temperature // text appearing in smokeview unit menu
1           // number of FDS variables that have temperature units
temp        // FDS variable, appears in colorbar heading
3 1         // number of units, default unit number
C           // first unit
1.0 0.0     // newunitval = 1.0*oldunitval + 0.0
F           // second unit
1.8 32.0    //
K           // third unit
1.0 273.15

UNIT
velocity    // text appearing in smokeview unit menu
4           // number of FDS variables that have velocity units
U-VEL       // first variable
V-VEL       // second variable
W-VEL       // third variable
Speed       // fourth variable
3 1         // number of units, default unit number
m/s
1.0 0.0
mph
2.236932 0.0
f/s
3.280833 0.0

```

**VECTORSKIP** Specifies what vectors to draw. For example, if this parameter is set to 2 then every 2nd vector is drawn when displaying vectors. (default: 1)

**VIEWPOINT** Specifies the internal Smokeview parameters used to record a scene's viewpoint and orientation. This parameter is set automatically by Smokeview when a .ini file is created. (default: none)

**XYZCLIP** Specifies clip planes in physical coordinates. There are six clipping planes, a minimum and maximum X, a minimum and maximum Y, a minimum and maximum Z. Each clipping plane may be used or not. The first parameter is 1 or 0 and specifies whether clipping is turned on or off. The next three lines specify clipping parameters for the X, Y and Z axes. Each line has the format

```
minflag min-clipval maxflag max-clipval
```

where the two flags, minflag and maxflag are 1 if turned on or 0 if turned off. Clipping is specified with the clipping dialog box found under the Options menu item. (default:

```

0
0 0.0 0 0.0
0 0.0 0 0.0
0 0.0 0 0.0

```

**ZOOM** Specifies the zoom amount used to display a Smokeview scene using two parameters, an integer zoom index and a floating point zoom amount. If the zoom index is 0-4 then the zoom amount is 0.25, 0.5, 1.0, 2.0 and 4.0 respectively. If the zoom index is negative then the second parameter is used to specify the zoom amount. (default: 0 1.0)

### D.1.6 Tour Parameters

**GLOBALTENSION** Specifies whether the tour spline tension parameter found in the dialog box should (1) or should not (0) be used for ALL tour keyframes. (default: 1)

**SHOWPATHNODES** Specifies whether the path nodes should (1) or should not (0) be drawn. This is a debugging parameter, not normally used. (default: 0)

**SHOWTOURROUTE** Specifies whether the tour route should (1) or should not (0) be drawn. (default: 0)

**TOURS** Keyword used to specify the tours. The format is

```
TOURS
ntours - number of tours
label for tour
nkeyframes - number of keyframes for first tour
time xpos ypos zpos 1 az elev bias continuity tension zoom localspeedflag
...
time xpos ... for last keyframe
nkeyframes for 2nd tour
...
...
```

If a cartesian view direction is specified then instead of 1 az elev above use 0 xview yview zview where xview, yview, zview are the coordinates of the view direction. The *Circle* tour is not stored in the .ini file unless it has been changed by the user. The tour entry created by using the **Add** button in the **Tour** dialog box is given by

```
TOURS
1
Added Tour 1
2
0.0 -1.0 -1.0 -1.0 0 0.0 0.0 0.0 0.0 0.0 1.0 0
100.0 7.4 9.0 7.4 0 0.0 0.0 0.0 0.0 0.0 1.0 0
```

**TOURCOLORS** Keyword used to specify the tour colors. The colors as before consist of a red, green and blue component ranging from 0.0 to 1.0 . One can override Smokeview's choice for the path, the path knots for both the selected and un-selected case. One may also specify the color of the time labels and the location of the object or avatar on the tour at the current time. The foreground color is used when a color component less than 0.0 is specified. Default:

```

TOURCOLORS
1.000000 0.000000 0.000000 :selected path line
1.000000 0.000000 0.000000 :selected path line knots
0.000000 1.000000 0.000000 :selected knot
-1.000000 -1.000000 -1.000000 :path line
-1.000000 -1.000000 -1.000000 :path knots
-1.000000 -1.000000 -1.000000 :text
1.000000 0.000000 0.000000 :avatar

```

**TOURCONSTANTVEL** Specifies whether the avatar should (1) or should not (0) traverse the path with a constant velocity. (default: 1)

**VIEWALLTOURS** Specifies whether all (1) tours should be drawn. (default: 0)

**VIEWTOURFROM** Specifies whether the scene should (1) or should not (0) be observed from the selected tour.

**VIEWTIMES** Specifies the tour start time, tour stop time and number of points to specify a tour. (default: 0.0 100.0 1000)

### D.1.7 Realistic Smoke Parameters

**ADJUSTALPHA** The ALPHA parameter in OpenGL/Smokeview is used to specify the transparency of an object. The ALPHA's are computed in FDS and adjusted in SMOKEVIEW according to value of the ADJUSTALPHA keyword. If ADJUSTALPHA is zero, then no adjustments to ALPHA are made. If ADJUSTALPHA is one, then ALPHA's are adjusted for non-orthogonal view directions (ALPHA is increased to account for longer path lengths along non-orthogonal view directions. If ADJUSTALPHA is two, then ALPHA's are adjusted as in the ADJUSTALPHA=1 case and are also set to zero on wall at blockage boundaries (this reduces graphical artifacts). (default: 1)

**FIRECOLOR** Specifies the color of the fire in red, green, blue coordinates. Each color component is an integer ranging from 0 to 255. (default: 255, 128, 0)

**FIREDEPTH** Specifies the depth at which the fire is 50 percent transparent. (default: 2.0)

**SMOKECULL** Cull (or do not draw) smoke if it is outside of the viewing frustum. (default: 1)

**SMOKESHADE** Grey level of smoke, may range from 0 to 255. (default: 255)

**SMOKESKIP** To speed up smoke drawing, spatial frames may be skipped. Allowable parameters are (0, 1, 2). (default: 0)

**SMOKETHICK** The ALPHA transparency parameter may be divided by a power of two ( $2^{\text{SMOKETHICK}}$ ) to make the smoke thinner. Parameters may range from 0 to 7. (default: 0)

### D.1.8 Zone Fire Modeling Parameters

**SHOWHZONE** Specifies whether upper layer temperatures should be (1) drawn horizontally or not (0). (default: 0)

**SHOWVZONE** Specifies whether upper layer temperatures should be (1) drawn vertically or not (0). (default: 1)

**SHOWHAZARDCOLORS** Specifies whether upper layer temperatures should be (1) drawn in terms of hazard or drawn in terms of a standard color scale (0). (default: 0)

## D.2 Smokeview Parameter Input File (.smv file)

The FDS software outputs simulation results into the Smokeview input file with extension `.smv` and various output data files whose format is documented in the next section. A `.smv` file is a formatted ascii text file consisting of a set of KEYWORDS followed by DATA describing the FDS case's geometry, data file names and contents, sensor information *etc.*

### D.2.1 Geometry Keywords

**GRID** This keyword specifies the number of grid cells in the X, Y and Z directions. For example,

```
GRID
10 20 30
```

specifies that there are 10, 20 and 30 grid cells in the X, Y and Z directions respectively.

**PDIM** This keywords specifies the physical length of the simulation domain in the X, Y and Z directions. For example,

```
PDIM
6.0 12.0 5.0
```

specifies that the length of the X, Y and Z directions is 6.0, 12.0 and 5.0 m respectively.

**SHOW\_OBST(HIDE\_OBST)** This keyword specifies when a blockage should be shown(hidden). For example,

```
SHOW_OBST 2
10 120.1
```

specifies that the tenth blockage in mesh 2 should be opened at 120.1 seconds. This keyword is automatically added to the `.smv` file by FDS.

**OBST** This keyword specifies internal blockages. A FORTRAN 90 code segment describing the format of OBST data is given by:

```
read(5,*)nb
do i = 1, nb
  read(5,*)x1(i),x2(i),y1(i),y2(i),z1(i),z2(i), id(i),
    ... s1(i),...,s6(i),tx(i),ty(i),tz(i)
end do
do i = 1, nb
```

Table D.1: Descriptions of parameters used by the Smokeview OBST keyword

Variable(s)	type	Description
nb	integer	number of blockages or entries for the OBST keyword
x1, x2 y1, y2 z1, z2	float	floating point blockage bounds
id	integer	blockage identifier
s1, s2 s3, s4 s5, s6	integer	index of surface (SURF) used to draw blockage sides
tx, ty, tz	float	texture origin
ib1, ib2 jb1, jb2 kb1, kb2	integer	Indices used to define blockage bounds in terms of grid locations.
colorindex	integer	Type of coloring used to color blockage. -1 - default color -2 - invisible -3 - use red, green, blue and alpha to follow (values follow) n>0 - use n'th color table entry
blocktype	integer	Defines how the blockage is drawn. 0 - regular block 2 - outline 3 - smooth
red, green, blue alpha	float	Each color value ranges from 0.0 to 1.0 . The alpha <i>color</i> represents transparency, alpha=0.0 is transparent, alpha=1.0 is opaque.

```

      read(5,*) ib1(i), ib2(i), jb1(i), jb2(i), kb1(i), kb2(i),
      ... colorindex(i) blocktype(i),
      ... red(i), green(i), blue(i), alpha(i)
end do

```

where the parameters are defined in Table D.1. The arrays `x1`, ..., `z2` and `ib1`, ..., `kb2` are required. All other arrays are optional and may be omitted.

**TRNX,TRNY,TRNZ** The `TRNX`, `TRNY`, `TRNZ` keywords specify grid nodes in the `X`, `Y`, `Z` coordinate directions. A FORTRAN 90 code segment describing the format of `TRNX` data is given by:

```

      read(5,*) nv
      do i = 1, nv
        read(5,*) idummy
      end do

```



```

do i = 1, nv
  read(5,*)xplt(i)
end do

```

TRNY and TRNZ data entries are defined similarly. The first `nx` data items are not required by Smokeview.

**VENT** The keyword specifies vent coordinates. A FORTRAN 90 code segment describing the format of VENT data is given by:

```

read(5,*)nv
do i = 1, nv
  read(5,*)xv1(i), xv2(i), yv1(i), yv2(i), zv1(i), zv2(i), id(i)
  ... sl(i), tx(i), ty(i), tz(i)
end do
do i = 1, nv
  read(5,*)iv1(i), iv2(i), jv1(i), jv2(i), kv1(i), kv2(i)
end do

```

where the parameters are defined in Table D.2. The arrays `xv1`, ..., `zv2` and `iv1`, ..., `kv2` are required. All other arrays are optional and may be omitted.

**OPEN\_VENT(CLOSE\_VENT)** This keyword specifies when a vent should be opened(closed). For example,

```

OPEN_VENT 2
3 15.6

```

specifies that the third vent in mesh 2 should be opened at 15.6 S.

**XYZ** The XYZ keyword defines the `.xyz` or `Plot3D` grid file name. A FORTRAN 90 code segment describing the format of XYZ data is given by:

```

read(5,"(a)")xyzfilename

```

where `xyzfilename` is a character variable containing the name of the `.xyz` file.

## D.2.2 File Keywords

**BNDF** The BNDF keyword defines the `.bdf` file name along with character labels used to describe the data contents of the boundary file.

**INPF** The INPF keyword specifies a file containing a copy of the FDS input file.

**ISO** The ISO keyword defines the `.iso` file name along with character labels used to describe the data contents of the isosurface file.

Table D.2: Descriptions of parameters used by the Smokeview VENT keyword

Variable(s)	type	Description
nv	integer	number of vents or entries for the VENT keyword
xv1, xv2 yv1,yv2 zv1,zv2	float	floating point bounds
id	integer	vent identifier
s1	integer	index of surface (SURF) used to draw vent
tx, ty, tz	float	texture origin
iv1, iv2 jv1, jv2 kv1, kv2	integer	Indices used to define vent bounds in terms of grid locations.
ventindex	integer	Type of coloring used to color vent. -1 - default color -99 of +99 -n of +n - use n'th palette color < 0 - do not draw boundary file over vent > 0 - draw boundary file over vent
venttype	integer	Defines how the vent is drawn. 0 - solid surface 2 - outline -2 - hidden
red, green, blue alpha	float	Each color value ranges from 0.0 to 1.0 . The alpha <i>color</i> represents transparency, alpha=0.0 is transparent, alpha=1.0 is opaque.

- PART** The `PART` keyword defines the `.part` file name along with character labels used to describe the data contents of the particle file.
- PL3D** The `PL3D` keyword defines the `.q` file name along with character labels used to describe the data contents for each Plot3D variable.
- SLCF** The `SLCF` keyword defines the `.sf` file name along with character labels used to describe the data contents of the slice file.

### D.2.3 Device (sensor) Keywords

**DEVICE** A *device* generalizes the notion of a sensor, sprinkler or heat detector. The `DEVICE` keyword defines the device location and name. This name is used to access the set of instructions for drawing the device. These instructions are contained in a file named `devices.svo`. The default location of this file on the PC is `C:\nist\fds\devices.svo`. This file may be customized by the user by adding instructions for devices of their own design (*i.e.* custom devices may be added to a Smokeview visualization without re-programming Smokeview). See ?? for more information on how to do this. The format for the `DEVICE` keyword is

```
DEVICE
device_name
x y z xn yn zn
```

where `device_name` is name of the entry in the object file Smokeview uses to draw the device and  $(x,y,z)$  and  $(xn,yn,zn)$  are the location and direction vector of the device.

**DEVICE\_ACT** The `DEVICE_ACT` keyword defines the activation time for a particular device. The format for the `DEVICE_ACT` keyword is

```
DEVICE_ACT
idevice time
```

where `time` is the activation time of the `idevice`'th device.

**HEAT** The `HEAT` keyword defines heat detector location data. A FORTRAN 90 code segment describing the format of `HEAT` data is given by:

```
read(5,*)nheat
do i = 1, nheat
  read(5,*)xheat(i),yheat(i),zheat(i)
end do
```

where `nheat` is the number of heat detectors and `xheat`, `yheat`, `zheat` are the  $x$ ,  $y$ ,  $z$  coordinates of the heat detectors.

**HEAT\_ACT** The `HEAT_ACT` keyword defines heat detector activation data. A FORTRAN 90 code segment describing the format of `HEAT_ACT` data is given by:

```
read(5,*)iheat, heat_time
```

where `heat_time` is the activation time of the `iheat`'th heat detector.

**SPRK** The `SPRK` keyword defines sprinkler location data. A FORTRAN 90 code segment describing the format of `SPRK` data is given by:

```
read(5,*)nsprink
do i = 1, nsprink
  read(5,*)xsprink(i),ysprink(i),zsprink(i)
end do
```

where `nsprink` is the number of sprinklers and `xsprink`, `ysprink`, `zsprink` are the `x`, `y`, `z` coordinates of the sprinklers.

**SPRK\_ACT** The `SPRK_ACT` keyword defines sprinkler activation data. A FORTRAN 90 code segment describing the format of `SPRK_ACT` data is given by:

```
read(5,*)isprink, sprink_time
```

where `sprink_time` is the activation time of the `isprink`'th sprinkler.

**THCP** The `THCP` keyword defines thermocouple location data. A FORTRAN 90 code segment describing the format of `THCP` data is given by:

```
read(5,*)ntherm
do i = 1, ntherm
  read(5,*)xtherm(i),ytherm(i),ztherm(i)
end do
```

where `ntherm` is the number of thermocouples and `xtherm`, `ytherm` and `ztherm` are the `x`, `y` and `z` coordinates of the thermocouples.

## D.2.4 Miscellaneous Keywords

**ENDIAN** The `ENDIAN` keyword defines whether a big or little endian<sup>4</sup> system is used to represent the data. The PC (both Windows and LINUX) uses the little endian system for representing numbers while most UNIX workstations use the big endian system. The PC version of Smokeview uses this keyword to determine whether data should be converted between these two systems before it is visualized. The valid `ENDIAN` parameters are `BIG` or `LITTLE`.

---

<sup>4</sup>Big-endian and little-endian are terms that describe the order in which a sequence of bytes are stored in computer memory. Big-endian is an order in which the *big end* (most significant value in the sequence) is stored first (at the lowest storage address). Little-endian is an order in which the *little end* (least significant value in the sequence) is stored first. The terms big-endian and little-endian are derived from the Lilliputians of Gulliver's Travels, whose major political issue was whether soft-boiled eggs should be opened on the big end or the little end.

**SYST**        The `SYST` keyword defines which system the FDS case was run on. The PC version of Smokeview uses this information (if the `ENDIAN` keyword is absent) to swap data bytes if the FDS case was run on a computer with a different *endian* format than the PC.

**TITLE1/TITLE2**    The `TITLE1` and `TITLE2` keywords allow one to specify extra information documenting a Smokeview case. These keywords and associated labels are added by hand to the `smokeview(.smv)` file using the format:

```
TITLE1
first line of descriptive text
TITLE2
second line of descriptive text
```

### D.3 Data File Formats (.s3d, .iso, .part, .sf, .bf, .q and CAD/.ge1 files)

Smokeview visualizes Fire Dynamics Simulator data using isosurface, particle, boundary, slice and Plot3D data files. Realistic geometric data created by DX2FDS is represented in a CAD or .ge1 file. This section documents the data format of each FDS data file by showing FORTRAN program code segments used to write the data out. All FDS files are unformatted and are opened using `OPEN` statements of the form:

```
OPEN (UNIT=IUNIT, FILE=FILENAME, FORM="UNFORMATTED")
```

where `IUNIT` is the FORTRAN unit number used for output and `FILENAME` is a character variable containing the file name. The CAD/.ge1 file is an ascii text file. In the following code segments the standard FORTRAN naming convention of associating variables beginning with I through N to integer variables and all other alphabetic characters to REAL variables is used. It is assumed that all variables take up four bytes. Variables having `DUMMY` in the name indicates that they are not used by Smokeview.

#### D.3.1 3D Smoke File Format

3D smoke files contain alpha values used by Smokeview to draw semi-transparent planes representing smoke, fire and water spray. The FDS software outputs 3D smoke frames at fixed time intervals. Note that char's are one byte, and "int's" and float's are four bytes. A *pseudo-code* representation of the 3D Smoke file is given by:

```
endian flag (int)
is1, is2, js1, js2, ks1, ks2 (6*int)
version (int)
for each time:
  time (float)
  chars_uncompressed, chars_compressed (2*int)
  compressed_data (chars_compressed*char)
end time
```

The endian flag is an integer one. Smokeview uses this number to determine whether the computer creating the 3D smoke file and the computer viewing the 3D smoke file use the same or different byte swap (endian) conventions for storing floating point numbers. The opacity data is compressed using run-length encoding.

### D.3.2 Isosurface File Format

Isosurface files are used to store one or more surfaces where each surface represents a region where the solution take one a particular value. The FDS software outputs isosurface frames at fixed time intervals. Note that char's are one byte, short's are two bytes and "int's" and float's are four bytes. A *pseudo-code* representation of the isosurface file is given by:

```
version                      (int)
len1,len2,len3              (3*int)
label1,label2,label3       ((len1+len2+len3+4)*char)
nlevels                     (int)
level_1, level_2, ..., level_nlevels (nlevels*float)
for each time:
  time                      (float)
  for each level
    nvertices               (int)
    ntriangles              (int)
    vertices_1, ..., vertex_nvertices (3*short*nvertices)
    triangles_1, ..., triangle_ntriangles (3*(byte/short/float)*ntriangles)
  end level
end time
```

The length of each `triangles_i` node is one byte if the number of triangles, `ntriangles`, is between zero and 255 (inclusive), two bytes if `ntriangles` is between 256 and 65536 (inclusive) and four bytes if `ntriangles` is greater than or equal to 65536. Note that the isosurface files are written using the C programming language. These files should be read the same way unless the programming language of choice supports binary I/O.

### D.3.3 Particle File Format

Particle files are used to store smoke/tracer particles and/or sprinkler droplet information. The particle file consists of a header and a series of particle frames. Each frame contains particle/droplet positions and data values for each particle in that frame. The FDS software outputs particle frames at fixed time intervals but there is no requirement by Smokeview for this to happen. Smokeview reads in the data one frame at a time noting the time read in for each frame. The header can be described using the code segment:

```
WRITE(IUNIT) DUMMY,DUMMY,DUMMY,IPART,NPPS
WRITE(IUNIT) IBAR,JBAR,KBAR
WRITE(IUNIT) (DUMMY,I=0,IBAR),(DUMMY,J=0,JBAR),(DUMMY,K=0,KBAR)
WRITE(IUNIT) NB
DO N=1,NB
  WRITE(IUNIT) IB1(N),IB2(N),JB1(N),JB2(N),KB1(N),KB2(N),1
END DO
WRITE(IUNIT) NV
DO N=1,NV
  WRITE(IUNIT) IV1(N),IV2(N),JV1(N),JV2(N),KV1(N),KV2(N),2
END DO
WRITE(IUNIT) NSPR
DO N=1,NSPR
  WRITE(IUNIT) XSP0(N),YSP0(N),ZSP0(N)
```

```
END DO
```

where DUMMY is written by FDS but not needed by Smokeview, IPART is the index of the scalar quantity associated with the particles, NPPS is the maximum number of particles per frame, IB1, IB2, etc are the indices of blocked grid cells, IV1, IV2, etc indicate vent cell nodes, and XSP0, YSP0, ZSP0 are the coordinates of the sprinklers. Grid coordinates are obtained by Smokeview from the Smokeview parameter input file. A particle frame is written using:

```
WRITE(IUNIT) TIME, NP, IDUMMY, (ISPR(N), N=1, NSPR)
WRITE(IUNIT) (XP(I), I=1, NP),
.           (YP(I), I=1, NP),
.           (ZP(I), I=1, NP),
.           (BP(I), I=1, NP)
IF (NASPR.GT.0) THEN
  WRITE(IUNIT) NSP
  WRITE(IUNIT) (XSP(I), I=1, NSP), (YSP(I), I=1, NSP), (ZSP(I), I=1, NSP)
ENDIF
```

where NP is the number of particles in this frame, ISPR denotes whether the sprinkler has activated, NSPR is the number of sprinklers, XP, YP, ZP are the particle coordinates, BP is the particle value, NASPR is the number of active sprinklers, NSP is the number of sprinkler droplets, and XSP, YSP, ZSP are the droplet coordinates.

### D.3.4 Slice File Format

Slice files are unformatted. The slice file header is written out using:

```
WRITE(IUNIT) CDUMMY
WRITE(IUNIT) CDUMMY
WRITE(IUNIT) CDUMMY
WRITE(IUNIT) I1, I2, J1, J2, K1, K2
```

where CDUMMY is a character strings of length 30. Smokeview obtains this information from the Smokeview parameter input file and not from the slice file. The sextuple (I1, I2, J1, J2, K1, K2) denotes the bounding grid cell nodes of the slice to be displayed. The sextuple indices correspond to grid cell nodes, or corners, thus the entire grid would be represented by the sextuple (0, IBAR, 0, JBAR, 0, KBAR). The grid node positions are obtained from the Smokeview parameter input file.

The FDS software outputs slice frames at fixed time intervals but there is no requirement by Smokeview for this to happen. Each slice frame is written using:

```
WRITE(IUNIT) TIME
WRITE(IUNIT) ((QQ(I, J, K), I=I1, I2), J=J1, J2), K=K1, K2)
```

where TIME is the time in seconds when the data is output and QQ are the data values.

### D.3.5 Terrain File Format

Terrain files are unformatted and are output by the Wildland-Urban interface (WUI) version of FDS. The terrain file header is written out using:

```

WRITE(IUNIT) 1
WRITE(IUNIT) VERSION
WRITE(IUNIT) XMIN, XMAX, YMIN, YMAX
WRITE(IUNIT) NX, NY
WRITE(IUNIT) (ZELEV(I, J), I=1, NX), J=1, NY)
WRITE(IUNIT) (STATE(I, J), I=1, NX), J=1, NY)

```

The first “1” is used by Smokeview to determine whether the file was written on a big or little endian computer. VERSION is an integer value representing the file version (initially 1), XMIN, XMAX, YMIN, YMAX are floating point values defining the terrain boundary, NX and NY are integers defining the number of cells in the  $x$  and  $y$  direction, ZELEV are floating point values defining the elevation at each grid cell and STATE are one byte integers defining the states at each grid cell.

The WUI FDS software outputs terrain frames at fixed time intervals but there is no requirement by Smokeview for this to happen. Because, cell states change infrequently, the cell states are only output when they change. Each terrain frame is written using:

```

WRITE(IUNIT) TIME
WRITE(IUNIT) NCHANGES
WRITE(IUNIT) (CELLINDEX(I), I=1, NCHANGES)
WRITE(IUNIT) (CELLSTATES(I), I=1, NCHANGES)

```

where TIME is the time in seconds when the data is output, CELLINDEX are the cell indices ( $cellindex = (J - 1) * NY + I - 1$ ) of the cells that have changed state and CELLSTATES are the value of the changed state. The data for the very first time step (at  $t = 0.0$ ) is not output explicitly but is assumed from information previously output in the header.

### D.3.6 Boundary Files

The boundary files are unformatted. The header is written out using:

```

WRITE(IUNIT) QUANTITY
WRITE(IUNIT) SHORT_NAME
WRITE(IUNIT) UNITS
WRITE(IUNIT) NPATCH
DO 10 I=1, NPATCH
    WRITE(IUNIT) IP1(I), IP2(I), JP1(I), JP2(I), KP1(I), KP2(I)
10 CONTINUE

```

where QUANTITY, SHORT\_NAME and UNITS are character strings of length 30, NPATCH is the number of planes (or *patches*) that make up the solid boundaries plus the external walls. The sextuple arrays (IP1, IP2, JP1, JP2, KP1, KP2) defines the cell nodes of each patch. The user does not prescribe these.

The FDS software outputs boundary frames at fixed time intervals but there is no requirement by Smokeview for this to happen. The boundary file frame is written using:

```

WRITE(IUNIT) TIME
DO 10 I=1, NPATCH
    WRITE(IUNIT) ((Q(I, J, K), I=IP1(I), IP2(I)), J=JP1(I), JP2(I)), K=KP1(I), KP2(I))
10 CONTINUE

```



### D.3.7 Plot3D Data

Field data is dumped periodically by FDS in a format used by the graphics package **Plot3D**[18]. The **Plot3D** data sets are single precision (32 bit reals), whole and unformatted. Note that there is blanking, that is, blocked out data points are not plotted. The grid data is written out to a file called `casename.xyz` using:

```
WRITE(IUNIT) IBAR+1, JBAR+1, KBAR+1
WRITE(IUNIT) ( ( (X(I), I=0, IBAR), J=0, JBAR), K=0, KBAR),
.             ( ( (Y(J), I=0, IBAR), J=0, JBAR), K=0, KBAR),
.             ( ( (Z(K), I=0, IBAR), J=0, JBAR), K=0, KBAR),
.             ( ( (IBLK(I, J, K), I=0, IBAR), J=0, JBAR), K=0, KBAR)
```

where `IBAR+1`, `JBAR+1`, `KBAR+1` are the number of grid cells along the `I`, `J`, `K` coordinate directions, `X`, `Y` and `Z` are the physical coordinates of the cell corners, and `IBLK` is an indicator of whether or not the cell is blocked. If the point `(X, Y, Z)` is completely embedded within a solid region, then `IBLK` is 0. Otherwise, `IBLK` is 1. The flow variables are written to a file using:

```
WRITE(IUNIT) IBAR+1, JBAR+1, KBAR+1
WRITE(IUNIT) ZERO, ZERO, ZERO, ZERO
WRITE(IUNIT) ( ( (QQ(I, J, K, N), I=0, IBAR), J=0, JBAR), K=0, KBAR), N=1, 5)
```

The five channels `N=1, 5` are by default the temperature (C), the  $u$ ,  $v$  and  $w$  components of the velocity (m/s), and the heat release rate per unit volume ( $\text{kW/m}^3$ ).

### D.3.8 CAD/GE1 file format

The program `DXF2FDS` (see <http://fire.nist.gov/fds/refs/tools.html#dxf2fds>) creates a FDS input file and a Smokeview geometric description file (.GE1) given a CAD description of the building being modelled. The CAD description must be in a *dxf* format and created using *3D face* commands. The .GE1 file has a simple text format and is described below.

```
[APPEARANCE]
nappearances
string (material description)
index r g b twidth, theight
tfile
:
:   The above entry is repeated nappearances-1 more times
:
[FACES]
nfaces
x1 y1 z1 x2 y2 z2 x3 y3 z3 x4 y4 z4 index
The above line is repeated nfaces-1 more times.
```

**nappearances**     Number of appearance entries to follow Each appearance entry has 3 lines.

**string**           A material description is written out by `DX2FDS` but is ignored by Smokeview.

**index**            An index number starting at 0.

**r, g, b** Red green and blue components of the CAD face used when a texture is not drawn. The values of r, g and b range from 0 to 255. If a color is not used then use -1 for each color component.

**twidth, theight** Textures are tiled or repeated. The characteristic width and length of the texture file is twidth and theight respectively.

**tfile** The name of the texture file. If one is not used or available then leave this line blank.

**nfaces** Number of face entries to follow. Each face entry has one line.

**x1/y1/z1/.../x4/y4/z4** x,y,z coordinates of a quadrilateral. The four corners of the quad must lie in a plane or weird effects may result when Smokeview draws it. (This is a requirement of OpenGL). The four points should be in counter-clockwise order.

**index** Points to a material in the [APPEARANCE] section.

## Appendix E

# Frequently Asked Questions

This appendix contains a list of frequently asked questions concerning Smokeview. An updated list may be found at <http://fire.nist.gov/smokeview> .

### E.1 How can I keep up with new releases and other information about FDS and Smokeview?

An e-mail list has been set up to alert users to changes, upgrades, *etc.* To subscribe to the list, send an email to [listproc@nist.gov](mailto:listproc@nist.gov) and put the line

```
subscribe fdsusers Jane Doe
```

in the text where Jane Doe is your name. Subject line is unimportant. To get off the list, send an email to the same address and write

```
unsubscribe fdsusers
```

Also, you can check the web page <http://fire.nist.gov/download> to see the date of the most recent update.

### E.2 Can I run Smokeview on other machines?

Smokeview is more complicated to compile on other platforms because it uses several freeware libraries that need to be compiled and installed on that particular machine. We have compiled Smokeview on the PC, SGI (IRIX), and Red Hat Linux, but with the exception of the PC executable, there's no guarantee that something we compile on one of our machines is going to work on yours. Worse comes to worse, use a PC to visualize the output of a calculation run on your other machine.

### E.3 Smokeview doesn't look right on my computer. What's wrong?

Some video cards do not use enough bits to represent the *depth buffer*. The depth buffer is used to determine when objects are hidden and therefore should not be drawn. A problem then can occur when the video card reduces the number of depth buffer bits from 32 to 16 in order to speed up its performance (less data to transport). This results in straight lines or polygonal edges appearing to be scalloped. Often reducing the acceleration setting for the video card can eliminate this problem. To do this:

- click your mouse on the desktop to bring up the "Display Properties" dialog box.
- click on the "Advanced..." button
- Find the dialog panel that lets you select the "Hardware acceleration". Reduce the acceleration (re-booting if necessary) until the scalloping disappears.

If you have upgraded your PC recently, make sure you have loaded the latest drivers for your video card.

## **E.4 How do I make a movie of a Smokeview animation?**

You make a movie of a Smokeview animation by converting the visualized scene into a series of PNG or JPEG files, one file for each time. More specifically:

- Set up Smokeview orienting the scene and loading the desired data files.
- Select the Options/Render menu and pick the desired frame rate. The more frames you include in the animation, the smoother it will look. Of course more frames results in larger file sizes.
- Use a program such as the Antechinus Media Editor or Platypus Animator available at <http://www.c-point.com>

## **E.5 Smokeview is running much slower than I expected. What can I do?**

A likely explanation is that your video drivers are not hardware accelerated for OpenGL. You need to download the latest drivers for your video card. You may also try playing around with the hardware acceleration settings (see E.3).

## **E.6 I loaded a particle file but Smokeview is not displaying particles. What is wrong?**

This is a bug in your video card. To see particles, reduce your hardware acceleration setting (see E.3).

## **E.7 Why do blockages not appear the way I defined them in my input file? ( invisible blockages are visible, smooth blockages are not smooth etc ) What is wrong?**

Blockage appearance is controlled by the Show/Hide  $\zeta$  Geometry  $\zeta$  Blockages menu setting. Blockages will appear as defined in the input file if you set Defined in Input File under this menu. To set the behavior to drawing blockages as defined in the input file, use the SHOWBLOCKS preference file keyword. The SHOWBLOCKS keyword defined in the smokeview.ini preference file ( or the casename.ini preference file for your case ) may be set to 1 to cause Smokeview to display blockages as defined in the input file. As always, you may also create a preference file, after you have set Smokeview menu options to display scenes as desired.