# SANDIA REPORT

SAND2005-1469
Unlimited Release
Printed March 2005

# Users Manual for CAFE-3D: A Computational Fluid Dynamics Fire Code

Ahti Suo-Anttila, Carlos Lopez, Imane Khalil

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico  87185 and Livermore, California  94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

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

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2005-1469
Unlimited Release
Printed March 2005

# Users Manual for CAFE-3D: A Computational Fluid Dynamics Fire Code

Ahti Suo-Anttila
Alion Science and Technology
6000 Uptown N.E. Suite 300
Albuquerque, NM 87110

Carlos Lopez and Imane Khalil
Sandia National Laboratories
Albuquerque, NM 87185

## Abstract

The Container Analysis Fire Environment (CAFE) computer code has been developed to model all relevant fire physics for predicting the thermal response of massive objects engulfed in large fires. It provides realistic fire thermal boundary conditions for use in design of radioactive material packages and in risk-based transportation studies. The CAFE code can be coupled to commercial finite-element codes such as MSC PATRAN/THERMAL and ANSYS. This coupled system of codes can be used to determine the internal thermal response of finite element models of packages to a range of fire environments. This document is a user manual describing how to use the three-dimensional version of CAFE, as well as a description of CAFE input and output parameters. Since this is a user manual, only a brief theoretical description of the equations and physical models is included.

# Acknowledgments

The authors wish to thank Alexander L. Brown and Stephen W. Webb from Sandia National Laboratories and Jorman A. Koski, retired from Sandia and currently working as a consultant to Sandia, for reviewing this document and providing many helpful suggestions. Funding for the development of the code and preparation of this manual was provided by the Office of Environmental Management and the Office of Civilian Radioactive Waste Management of the Department of Energy.

# Contents

# Figures

# Tables

This page intentionally left blank.

# Executive Summary

The Container Analysis Fire Environment (CAFE) computer code has been developed to model all relevant fire physics for predicting the thermal response of massive objects engulfed in large fires. It provides realistic fire thermal boundary conditions for use in design of radioactive material packages and risk-based transportation studies. The CAFE code can be coupled with finite-element models of specific package designs. This coupled system can be used to determine the internal thermal response of engulfed objects to a range of fire environments. The code is designed to run rapidly on desktop workstations, typically yielding results of suitable accuracy in several hours.

This document is a user manual describing how to use the three-dimensional version of CAFE, as well as a description of various input and output parameters. The input description contains actual examples from CAFE input files. Each variable and the required parameters are described. This document includes many variables that may never be used in a typical CAFE run, but could be used in more general computational fluid dynamics applications.

This page intentionally left blank.

# Nomenclature

| | |
|---|---|
| CAFE | Container Analysis Fire Environment |
| CFD | computational fluid dynamics |
| BC | boundary conditions |
| CPM | CAFE-3D Porosity Method |
| EOS | equation of state |
| FAVOR$^{\text{TM}}$ | Fractional Area and Volume Object Representation method |
| FE | finite element |
| GMV | General Mesh Viewer |
| LES | Large Eddy Simulation |
| RANS | Reynolds Averaged Navier Stokes |
| UID | User ID |
| URANS | Unsteady Reynolds Averaged Navier-Stokes |
| VLES | Very Large Eddy Simulation |

This page intentionally left blank.

# 1. Introduction

The CAFE code (Suo-Anttila et al., 1999 and Lopez et al., 2003) is a computational fluid dynamics fire model that includes all of the dominant physics present in fires. It is intended to link with commercial finite element codes supplying realistic fire environment boundary conditions to finite element (FE) models of packages. CAFE was designed to run rapidly so that it can be used on desktop workstations. It is written in FORTRAN-77, and can be compiled and run on any machine with an F77 compiler. Currently, CAFE is coupled to the commercial MSC PATRAN/THERMAL (P/THERMAL) suite of analysis codes. A beta version of CAFE coupled to the ANSYS code has recently been completed, and is undergoing testing. CAFE is written such that it can be linked to other finite element analysis codes that have the capability of importing user-supplied subroutines.

This document contains a description of CAFE features, a brief theory section, a description of input and output, and an example problem. The theory section is brief because this is a user manual, dedicated to code features and input/output descriptions, not a theory manual.

This page intentionally left blank.

# 2. CAFE Development History

The CAFE code has been in development for a number of years. The earliest versions of CAFE were one-dimensional correlations that could be applied to the surface of a cylinder, or its end plates. These correlations were derived either from experiment, or analytically, and were intended to be a first order improvement over a constant temperature boundary condition that is often used to model large pool fires.

Although the initial CAFE correlations applied variable boundary conditions to the surface of a finite element model of a package, such as a cylinder, they did not include many important effects that are present in real fires. For example, a fire has an internal temperature distribution that is not uniform. Thus, any object will be heated non-uniformly depending upon its location within the fire. In addition wind will tilt and stretch a fire such that the global temperature distribution is altered from the no-wind condition. A large object placed within the fire will absorb heat from the fire and cause flames to be diverted around the object, thus altering the size, shape, and temperature of the fire. Such effects cannot be accounted for by simple correlations. Instead, a physical model, such as a computational fluid dynamics (CFD) code is required.

To extend the utility of the CAFE code a two-dimensional fire model was developed. The first 2-D model utilized a vorticity-velocity formulation to solve the momentum equations and resulted in a code that was fast, compact and efficient at applying engulfing fire boundary conditions. The main drawback of the vorticity-velocity approach was the inability to extend the approach to complex internal and external boundary conditions which are required to model open pool fires with engulfed objects and external conditions such as wind. Rather than continuing the development of the vorticity-velocity formulation, existing source code from the Isis-2D CFD code was adopted. The Isis-2D source code was available, partially validated for use in fires, and could be applied directly to satisfy many of the requirements of the CAFE-2D code. Thus, the two-dimensional version of CAFE was built from the Isis-2D CFD code.

The 2-D CAFE code was capable of modeling many hypothetical situations, however it required the user to model a three dimensional object using a combination of 2-D cutting planes. That approach is an art form and requires the user to be familiar with the nuances of 2-D versus 3-D CFD effects. Large calorimeter experimental data (Kramer et al., 1998) demonstrated that significant 3-D effects were present, even with light winds. Those effects could not be accounted for a-priori by a 2-D CAFE code. For example, although the cylinder was located at least 1 meter inside the edges of the fuel pool, a considerable amount of fire necking and tilting occurred, such that a fraction of the cylinder (the end plates) were outside the flame zone. These fire necking and tilting effects can only be accounted for automatically by extending the CAFE code to three dimensions.

The extension of the CAFE code to 3-D expands the capabilities of CAFE such that almost any fire and object configuration can be modeled without requiring the user to artificially introduce known experimental effects, such as flame necking. Building a 3-D CFD code from scratch, or even extending the 2-D CAFE code to 3 dimensions was a task that was well beyond the CAFE development budget. Instead, the Isis-3D code was adopted as the finite volume CFD engine for

CAFE-3D. Thus CAFE-3D development consists of adapting and modifying the coupling routines between the FE model and the finite volume CFD code Isis-3D.

Recent efforts in the validation and calibration of CAFE by comparison of the code to experimental data is discussed in Section 4.9.

# 3.  CAFE-3D Code Description

CAFE-3D consists of three linked codes, P/THERMAL, Isis-3D, and ulib.a.  P/THERMAL is the FE solver for the solid body model.  Isis-3D is the finite volume CFD code that supplies the fire heat flux boundary conditions to the object.  Ulib.a is a user library that comprises the linking subroutines, which passes geometric and thermal data between the FE and CFD codes. Another application that is not included as part of CAFE, but forms an integral part of any analysis is PATRAN, which is the geometric modeler, mesh generator, and pre-and post processor for the P/THERMAL finite element model.

In this report the following terms (code names) will apply, **CAFE** (or CAFE-3D) will refer to the linked system of codes.  Isis-3D is the CFD code which is used in both standalone and linked applications.  In order to avoid code name confusion, the Isis-3D code will be referred to as **standalone CAFE**.  Ulib.a refers to those subroutines that control the flow of information between P/THERMAL and Isis-3D.  PATRAN is the pre- and post-processor that is used for building P/THERMAL models and viewing output.  There are additional graphics routines that can be used for viewing output, such as Xplot, GMV, and Geomview.  These three applications are not required to run CAFE but are very helpful to the user for error checking, viewing fire behavior, and understanding the results that are produced.

The CFD model in CAFE is coupled to the finite element code P/THERMAL, through a set of user-defined subroutines in ulib.a.  In principle, CAFE can be coupled to any commercial finite-element code that allows users to supply their own subroutines for applying thermal boundary conditions.  Surface temperature information of an object is passed from the finite-element model by the ulib.a subroutines to the CFD code, which then uses that information to predict a position-dependent heat-flux. The heat-flux is passed back to P/THERMAL, which uses the heat-fluxes as local boundary conditions for predicting the object internal thermal response. Thus, P/THERMAL has time- and position-dependent fire boundary conditions that depend upon the physics of the problem. The heat-flux and temperature data passing between the CFD and FE codes is user-controlled. The control logic is based upon the surface temperature rise of the FE object(s) in the model. The user assigns a minimum temperature rise (5K is a typical value). For each FE time step, the CAFE code checks if any surface node has risen more than the user-assigned value. If a sufficient temperature change has occurred anywhere on the object(s) surface(s) then the CAFE CFD model predicts a new distribution of surface heat-fluxes. In generating a surface heat-flux distribution, CAFE integrates the equations of motion and energy forward in time until the new object surface temperature distribution is in quasi-steady equilibrium with the fire.  The process of calling CAFE-3D repeatedly allows for puffing and the variable nature of fire dynamics to be taken into account in the heat transfer calculation.

Visualization of the output from CAFE can be done by various means, depending upon the needs of the user.  The most obvious visualization is PATRAN, which will allow the user to view the predicted surface temperature of the package.  However, PATRAN will not allow viewing of the fire and smoke behavior. To view the fire the user has several choices. CAFE will output several different file formats that can be read by either commercial or government visualization software. Details on the types of output files, and the associated visualization software, can be found in Chapter 8 of this document.

This page intentionally left blank.

# 4. CAFE Numerical and Physical Models

Since CAFE is a linked system of three codes, the numerical and physical models in each code apply to the overall linked system of codes. The numerical and physical models described in this chapter apply to the CAFE-3D standalone code, which is the finite volume CFD code that supplies surface heat flux boundary conditions to the solid object. The numerical models and methods in P/THERMAL can be obtained from the relevant PATRAN manuals from MSC Corporation.

A brief description of the numerical and physical models in CAFE follows. The description is terse because this is a user manual, dedicated to describing features, and how to use the code, not a theory manual.

## 4.1 CAFE-3D Numerical and Physical Models

Some of the numerical and physical models included in CAFE-3D are as follows. The governing equations are: three momentum equations for predicting the 3-D velocity/momentum field, a mass continuity equation, an energy equation for predicting the temperature field, an equation of state, and an indefinite number of scalar transport equations for tracking the flow of fuel vapor, oxygen, and soot, other species of interest, and turbulent kinetic energy if a one-equation turbulence formulation is chosen. CAFE-3D uses a finite volume approach with orthogonal Cartesian coordinates for discretizing the governing equations. The use of orthogonal coordinates makes the discretization very similar to a finite difference method. In a finite volume formulation all vector quantities, such as heat-flux and momentum, are defined at the cell interfaces whereas scalar variables, such as temperature and pressure, are defined at cell centers. CAFE-3D uses a variable density version of the PISO (Issa, 1985) pressure-based solution algorithm to solve the flow equations (Suo-Anttila, 1993). Turbulence can be modeled by the Prandtl (Schlichting, 1979) one equation, a large eddy simulation (LES) or, more likely, very large eddy (VLES) formulation (Smagorski 1963, Ferziger, 1993, McDonough 2004), or other algebraic formulations. Diffusive thermal radiation transport is modeled with the Rosseland approximation. The rate of combustion is modeled by a combination of Arrhenius and eddy breakup models for the reaction rate of fuel and oxygen.

To model curved objects, such as cylinders or spheres, CAFE-3D uses a porosity method similar in form and function to the Fractional Area and Volume Object Representation method (FAVOR$^{TM}$) (Hirt, et al. 1985). For a detailed discussion of FAVOR see the web site http://flow3d.com. The CAFE-3D Porosity Method (CPM) is a body fitted coordinate method that can be applied in an orthogonal Cartesian mesh. CPM models a curved surface by admitting a flat diagonal surface within a hexahedral finite-volume computational cell. Multiple cells with polygonal cross sections are used to represent a curved object such as a cylinder. CPM results in a much better representation of curved surfaces than a typical stair-step representation, which is often used with Cartesian coordinate systems. See Figure 1 for a comparison of the CAFE-3D Porosity Method with a conventional stair-step representation of a right circular cylinder. A segmented object representation, such as produced by CPM, has an equivalent heat transfer and flow representation as a finite-element CFD method would, but it does not incur the overhead of increased processor and memory requirements associated with either finite-element methods, or structured body fitted methods. With CPM, the flow areas on all computational cell surfaces are adjusted to account for the diagonal solid surface. That is, each face of a computational cell has both a solid and fluid area fraction, and the effect of these area fractions is included in the conservation equations solvers. Heat transfer between the gas and solid in the same computational cell is calculated by solving two energy equations. The energy equations use the surface area of the imbedded diagonal surface in calculating the energy transfer rate.

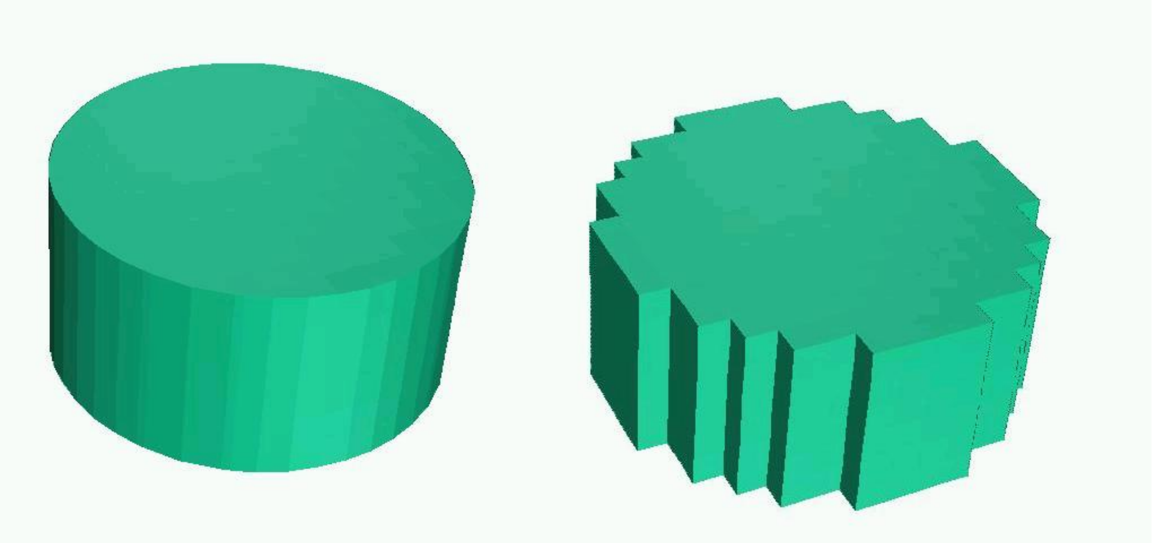

**Figure 1. A comparison of a cylinder using the CAFE-3D Porosity Method (CPM) (left) and the conventional stair step representation (right) often used in finite difference methods. Both objects have the same grid structure.**

## 4.2 Governing Equations in CAFE-3D

The governing equations in CAFE-3D are: three momentum equations, a continuity equation, the species equations, and one or two energy equations. A set of constitutive equations is included to couple the governing equations together. The constitutive equations are: a pressure equation, a fluid equation of state (EOS), and a set of chemical reaction equations. CAFE 3D also includes a set of one-dimensional subgrids, which use the same set of governing and constitutive equations. The 3-D and 1-D subgrids are coupled together by mass, energy, and species source terms evaluated at the beginning of each time step.

The primitive variables in CAFE-3D are the three momentum fluxes (product of density and velocity), temperature (energy equation), and mass fraction (species equations). The primitive variable, momentum flux, has the advantage of eliminating density interpolation error when using a pressure equation solution algorithm. It also has the advantage of a better variable density implementation, which is important when solving problems with large density variations such as fires.

The equations of motion that are solved simultaneously in CAFE-3D are the momentum equations, the continuity equation, the pressure correction equation, and the momentum correction equation.

The CAFE-3D momentum equation (Equation 1) is solved in its conservation form for the momentum flux vector (ρ**u),** where ρ is the gas density, **u** is the gas velocity vector, P is the fluid pressure, F is a drag coefficient, and μ is the gas viscosity, ε is the eddy diffusivity, and **g** is gravity. The area weighting effect of variable porosity due to the presence of structure (CPM) and/or an r-z coordinate system is included in the $\nabla$ operator in all the equations which follow. Expanded forms of these vector (and scalar) equations can be found in various textbooks [Bird et al. 1960].

$$\frac{\partial \rho \mathbf{u}}{\partial \mathrm{t}} + [\nabla \bullet \mathbf{u}\, \rho \mathbf{u}] = -\, \nabla \mathrm{P} - \rho \mathbf{u} \mathrm{F} + \nabla(\mu + \rho\varepsilon\mathbf{)}\, \nabla \mathbf{u} + \rho \mathbf{g} \qquad (1)$$

The effect of turbulent shear is only approximated in the momentum equation, by adding the molecular and eddy diffusivity in the viscous drag term. The full tensor form of turbulent eddy shear stress is not used because the dominant terms are included in the form shown above. The same turbulence simplification is made for the energy and species equations. Additional discussion of turbulence appears in Section 4.7.

The incompressible continuity equation in CAFE-3D is

$$\nabla \bullet \rho \mathbf{u} = 0 \;\; . \qquad (2)$$

The method of obtaining a simultaneous solution to both equations is described below.

The first step in solving the set of equations described above is to get an estimate of the momentum distribution (ρ$\mathbf{u}^*$), which is obtained by solving the momentum equations using the beginning of time step pressure distribution as a source term

$$\frac{\partial \rho \mathbf{u}^*}{\partial \mathrm{t}} + \left\{\nabla \bullet \mathrm{u}^{\mathrm{n}}\, \rho \mathbf{u}^*\right\} = -\nabla \mathrm{P}^{\mathrm{n}} - \rho \mathbf{u}^{\mathrm{n}} \mathrm{F} - \nabla(\mu + \rho\varepsilon)\nabla \mathbf{u}^* + \rho \mathbf{g}\, . \qquad (3)$$

The superscript * refers to an estimate of the momentum flux, and the superscript n refers to the beginning of the time step value.

The next step is to solve the continuity equation to obtain the mass divergence, which is used as a source term in the pressure corrector equation:

$$\frac{\partial \rho^*}{\partial \mathrm{t}} = -\, \nabla \bullet \rho \mathbf{u}^*\, . \qquad (4)$$

The pressure correction equation is derived from the momentum and continuity equations. Since only a correction to the momentum field is needed, the discretized form of the predictor momentum equation is subtracted from the end-of-time-step momentum equation resulting in a ‘momentum correction equation.’ The pressure correction equation is created by taking the dot product of the ∇ operator and the momentum correction equation, and then substituting in the corrector continuity equation. The result of these operations is:

$$\nabla^2 P^{n+1} = \left( \frac{\rho * - \rho^{n+1}}{\Delta t^2} \right) - \frac{\nabla \bullet u\left(\rho * - \rho^{n+1}\right)}{\Delta t} + \nabla^2 P^n \,. \tag{5}$$

Here, n+1 refers to the end-of-time step value, n is the beginning of time step value, * is the end of time step predictor value, and Δt is the time step size. The end of time step density appears in this equation and that is obtained from the assumption of uniform (or hydrostatic) pressure everywhere in the computational mesh combined with the equation of state. The solution of the pressure correction equation is used with the momentum correction equation (e.g., Equation 6) to obtain the final end-of-time-step momentum field:

$$\frac{\left((\rho \mathbf{u}^{n+1} - \rho \mathbf{u}*)\right)}{\Delta t} = -\nabla\left(P^{n+1} - P^n\right) + \frac{\mathbf{u}\left((\rho^{n+1} - \rho*)\right)}{\Delta t}. \tag{6}$$

Once at the end of time-step and the momentum field has been obtained, all of the transport equations for energy, species, and turbulence can be updated for their end of time step values.

Summarizing, the method of solution in CAFE-3D is:

1. Solve the momentum equations to obtain the predictor momentum distribution.
2. Integrate the continuity equation to obtain the predictor density distribution.
3. Solve the pressure equation for a new pressure field that satisfies the continuity equation.
4. Solve the momentum corrector equations for the corrected momentum field.
5. Solve the transport equations for the end of time step scalar variables.
6. Solve the energy equation for the end of time step temperature.
7. Solve the structure energy equation for the end of time-step structure (solid) temperature.

This sequence of calculations is then repeated for the next time step. Note that this solution algorithm does not require iteration for solution stability. Stability, and solution accuracy, is achieved by time step control rather than by iteration. In particular, the time step is regulated to keep the mass divergence error, and other key variables (e.g., EOS error, courant number, fluid temperature change, density change, eddy viscosity, or wall temperature change) below specified values. The accuracy of this approach has been evaluated and is discussed further in Section 4.9.

## 4.3 The Transport Equations

A set of transport equations is solved for the various chemical species that are present. The transport equations for chemical species are solved in their conservative form where ρ is the fluid density, $m_i$ is the mass fraction, **u** is the velocity vector, $D_{i,j}$ is the diffusion coefficient of species i thorough the bulk fluid j, and $S_i$ is a species source term. The diffusion coefficient is the sum of the molecular diffusion coefficient and the eddy diffusivity due to turbulence.

$$\frac{\partial \rho m_i}{\partial t} + \nabla \rho \mathbf{u} m_i = \nabla(\rho D_{i,j} + \rho\varepsilon)\nabla m_i + S_i \tag{7}$$

## 4.4 The Energy Equation

The CAFE-3D gas energy equation is:

$$\begin{aligned}\frac{\partial \rho c_p T}{\partial t} + \nabla \bullet \rho c_p \mathbf{u} T = {} & \nabla \bullet (k + \rho c_p \varepsilon)\nabla T + \Sigma\, h_s \frac{A_s}{V}(T_s - T\ ) + Q_v \\ & + \frac{1}{\gamma}\frac{DP}{Dt} - \frac{1}{2}\left[\frac{\partial}{\partial t}(\rho u^2) + \nabla \bullet \rho \mathbf{u} u^2\right]\end{aligned} \quad . \tag{8}$$

where $c_p$ is the specific heat at constant pressure, T is the gas temperature, $T_s$ is the temperature of structure s, a is an area weighting term that accounts for porous structure (CPM), k is the fluid conductivity (including diffusive radiation conductivity), ε is the eddy diffusivity, $h_s$ is a heat transfer coefficient to structure surface s, $A_s$ is the heat transfer surface area of structure s, V is the flow volume, and $Q_v$ is the volumetric heat source, γ is the ratio of specific heats. Heat sources due to radiation transport boundary conditions, chemical reactions, energy exchange with 1-D subgrids, and other explicit heat sources are lumped into the $Q_v$ term.

The last term in Equation 8, (i.e., $-\frac{1}{2}\left[\frac{\partial}{\partial t}(\rho u^2) + \nabla \bullet \rho \mathbf{u} u^2\right]$), is only applicable when the compressible option is turned on. Although this term is negligible in the incompressible solver, the same energy equation is used for both compressible and incompressible flow.

Radiation heat transfer is included in the energy equation as source, or sink terms. Radiation heat transfer is discussed further in Section 4.7.

The CAFE-3D structure energy equation is:

$$\rho_s c_s \frac{\partial T_s}{\partial t} = \frac{1}{a}\nabla \bullet k_s\ a\nabla T_s + \Sigma\, h_g \frac{A_g}{V}(T\ \ - T_s) + Q_s\ , \tag{9}$$

where $\rho_s$ $c_s$ is the structure density times specific heat, $k_s$ is the conductivity, $h_g$ is a heat transfer coefficient to gas in cell k, $A_g$ is the surface area for heat transfer to gas in cell g, V is the volume of the structure, and $Q_s$ is the volumetric heat generation in the structure. The summation refers to heat transfer with the gas in all of the adjacent cells (left, right, top, and bottom) and to the gas within the same cell.

## 4.5 Turbulence Models

When examining the literature there are numerous turbulence models to choose from. Unfortunately, none of these models provides an adequate model of turbulence in all cases. Direct numerical simulation of all turbulence eddy scales is not feasible because of the prohibitively large number of computational cells required. Other approaches such a Large Eddy Simulation (LES) and various Reynolds Averaged Navier Stokes (RANS) methods also have their deficiencies (McDonough 2004). Many people have claimed success in comparisons to experiment, however these are only for selected cases. Nevertheless some of these methods have been implemented in CAFE.

CAFE-3D has several different turbulence models to choose from. All are based on an eddy diffusivity concept. There are several algebraic models wherein the eddy diffusivity is a function of Reynolds number, or other parameters, and position. A Very Large Eddy Model, a one equation Unsteady Reynolds Averaged Navier-Stokes (URANS) model. The standard Two-Equation k-ε (RANS) model is not available.

The precise form of the algebraic turbulence models is discussed in the input section of this report. When using these algebraic models the user would specify proportionality constants as a function of position and fluid velocity, to create an estimate of the eddy diffusivity.

The LES model in CAFE-3D is based upon the concepts of Smagorinsky (1963). In a conventional zero-equation LES model, the eddy viscosity is calculated by the following formula:

$$\mu_{LES} = \max\left(\mu \ , \rho (C_s \Delta)^2 |S|\right), \tag{10}$$

where $C_s$ is a constant of proportionality (typically 0.15 - 0.2), $\mu$ is the fluid viscosity, $\Delta$ is the length scale of the turbulence (which in a zero equation model the cube root of the cell volume), and |S| is the magnitude of the deformation tensor:

$$|S|^2 = 2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + 2\left(\frac{\partial w}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2 . \tag{11}$$

However in order to use a LES model appropriately one must select a grid structure such that significant fraction of the turbulent kinetic energy resides in the resolved turbulent scales. Estimates are, that for LES modeling, this would require Reynolds number squared computational cells. For a typical 5-meter fire, LES would require many millions of computational cells. This is well beyond desktop workstation capability, which is what CAFE is designed for. In fact, applying LES to a 5-meter pool fire is barely within the capability of the fastest super computers. Further it is known that the Smagorinsky zero equation approach is valid only in homogeneous turbulence, not anisotropic turbulence found in fires. To include anisotropy, $C_s$ would no longer be a constant—instead it would vary as a function of time and position.

Very Large Eddy Simulation (VLES), approaches have been developed wherein fewer computational cells are used than in LES, and the subgrid turbulence scales are modeled by other methods. Although these methods show some merit, the coarse meshes typically employed adds numerical dissipation to the solution. This numerical dissipation coexists with the modeled turbulent dissipation and often results in too much dissipation. When using CAFE, coarse meshes are often employed, which by their nature include numerical dissipation. Since none of the turbulence models (LES, VLES, RANS, URANS) are sufficient for all cases, one is left with a difficult choice on how to proceed. It would appear that not including turbulence at all would result in fire calculations that have the semi-correct behavior since numerical dissipation would simulate the turbulent dissipation that an explicit turbulence model would provide. In fact, many sample CAFE calculations have demonstrated this to be the case. The difficulty lies in including an eddy breakup combustion model, which relies on turbulence models for calculating the rate of combustion. In CAFE this difficulty is overcome by including a combined Arrhenius and Eddy breakup combustion model, which is considerably less sensitive to the turbulence model employed.

In CAFE, large scale turbulence, such as puffing in fires, is automatically calculated provided that the mesh structure is fine enough. It is recommended that the user select the VLES option, which uses a position-dependent algebraic formula for the turbulent length scale. Further, the proportionality constants should be set to a low value so as not to add too much (i.e., negligible additional) dissipation to the solution, since numerical dissipation from a coarse mesh is unavoidable and unfortunately included. Finally, turbulence is used in CAFE primarily to scale the eddy breakup time in the combustion model. Since the turbulent eddy diffusivity is scaled by user input constants, the eddy breakup proportionality constants are adjusted to obtain good agreement with experiment. If this sounds like a parametric approach, it is. Comparisons with experiment have shown that a single set of turbulence/eddy breakup parameters can match data from several different fires, of various sizes and wind conditions. Thus, additional calibration of the coefficients may not be required. The current set of turbulent/eddy-breakup coefficients comprise the default CAFE-Fire model. Future development of CAFE will reduce the parametric nature of this approach by including a scale length dependence directly from a turbulence model.

Currently numerical dissipation is compensated for as follows: An estimate of the numerical dissipation based upon the users grid size and the local velocity is subtracted from the user selected model based turbulent dissipation. If the remainder from the subtraction is positive, the difference is added to the solution as the net eddy viscosity. This yields a close approximation to the actual turbulent dissipation level predicted by the turbulence model, without adding additional numerical dissipation (provided the user's mesh is small enough to satisfy the criterion).

Promraning and Rutland (2002), have indicated good agreement with experiment by combining a one equation turbulent kinetic energy transport model (described below) with a VLES model, wherein the length scale was proportional to the cell size. In their calculation a large number of cells were employed, but fewer than required by a LES simulation. This type of approach is possible in CAFE by turning on both the VLES and the one-equation turbulence transport model. However, if coarse grids are employed this approach may be too dissipative, and therefore not recommended.

A transient RANS One-Equation turbulence model is available in CAFE. This model uses a transport equation for the turbulent kinetic energy. The transport equation is solved in its conservative form for the kinetic energy of turbulence, as shown in the equation below:

$$\frac{\partial \rho k}{\partial t} + \nabla \rho \mathbf{u} k + \nabla\left(\mu + \frac{\varepsilon}{\rho}\right)\nabla k + \frac{\varepsilon}{\rho}\left(\nabla \times \mathbf{u}\right) - \frac{C_d \ \rho \ k^{1.5}}{l} = 0 \, , \tag{12}$$

where $\rho$ is the fluid density, k is the turbulent kinetic energy, $\varepsilon$ is the eddy diffusivity, and $C_d$=0.164 if *l* is the ordinary mixing length (Anderson et al. 1984). The mixing length *l* in the one-equation turbulence model is a user input algebraic function of position, or alternately proportional to cell size.

The relationship between turbulent kinetic energy and eddy diffusivity is

$$\varepsilon = C_k \, l \, (\, \kappa)^{½}, \tag{13}$$

where $C_k$ is a user input constant, typically 0.1. The last term in the equation is the turbulence dissipation rate. Note that in this one equation formulation, eddy dissipation is not a separately transported scalar as in conventional two-equation k-ε model, but is transported along with the turbulent kinetic energy and is a simple power law function of that energy. The use of a RANS model in a transient calculation has been criticized by many in the turbulence modeling field because RANS assumptions are based on steady flow turbulence. Nevertheless, some claim good agreement with experiment using Unsteady RANS methods (Promraning and Rutland 2002). As a final note, using the one equation turbulence model will use one transport equation in the code. If the user's model is using all of the available species (typically set to five), then the code may have to be recompiled with one additional transport equation before the one-equation turbulence model can be activated.

## 4.5 Combustion Chemistry and Soot Model

This section describes four separate reactions that CAFE-3D uses to model the chemical processes within hydrocarbon pool fire, such as JP8. This model is a variant of a turbulent flame model developed by Said et al. (1997). The relevant species in this model are the hydrocarbon fuel vapor (denoted F), atmospheric oxygen ($O_2$), solid carbon soot (C), an intermediate species (IS), and products of complete combustion (PC, which consists of $CO_2$ and $H_2O$ vapor). The identity of the intermediate species is unknown but not required for the model. The radiative properties of both the fuel F and intermediate species IS are assumed to be the same as methane.

The first model reaction is incomplete fuel combustion that produces carbon soot in addition to the products of complete combustion. The Combustion Soot Parameter $S_1$ describes the mass of soot produced per unit mass of fuel consumed in this reaction. The mass-based functional formula for this reaction is:

$$1 \, kg \text{ F} + (3.0\text{-}2.6S_1) \, kg \text{ O}_2 \rightarrow S_1 \, kg \text{ C} + (4.0\text{-}3.6S_1) \, kg \text{ PC} + (42\text{-}32S_1) \, MJ. \tag{14}$$

When $S_1$ = 0 no soot is produced and the only products are $CO_2$ and $H_2O$. As $S_1$ increases, soot production from this reaction increases, the amount of normal products of combustion decreases, and the heat of reaction goes down. One of the differences between the current combustion chemistry model and one presented by Said et al. is that they assumed that when fuel reacts with oxygen it does not produce soot ($S_1$ = 0). Recent comparisons with the data of Murphy and Shaddix (2004) show that a value of $S_1$ = 0.125 results in good agreement (Lopez et al., 2004).

The second model reaction is endothermic fuel cracking. This reaction is anaerobic and produces soot and intermediate species. The Cracking Soot Parameter $S_2$ describes the mass of soot this reaction produces per mass of fuel it consumes as follows:

$$1\ kg\ \mathrm{F} + 0.3\ MJ \rightarrow \mathrm{S}_2\ kg\ \mathrm{C} + (1\text{-}\mathrm{S}_2)\ kg\ \mathrm{IS}. \tag{15}$$

In this reaction, 0.3 *MJ* of heat is consumed for every kilogram of fuel that it reacts. Comparison to the data of Murphy and Shaddix suggests a value of S2=0.05 leads to good agreement with their data (Lopez et al., 2004).

The third reaction is carbon soot combustion. The chemical formula for this reaction is:

$$1\ kg\ \mathrm{C} + 2.6\ kg\ \mathrm{O}_2 \rightarrow 3.6\ kg\ \mathrm{CO}_2 + 32\ MJ. \tag{16}$$

The final model reaction is combustion of the intermediate species:

$$1\ kg\ \mathrm{IS} + \frac{3-2.6S_2}{1-S_2}kg\ \ \mathrm{O}_2 \rightarrow \frac{4-3.6S_2}{1-S_2}kg\ \ \mathrm{PC} + \frac{41.7-32S_2}{1-S_2}MJ\ . \tag{17}$$

The coefficients in this reaction were determined such that complete combustion of the intermediate species and soot produces the same species and thermal energy as direct combustion of the fuel.

The Arrhenius rate model is used for all four reactions. As such, the consumption of fuel, soot, and intermediate species are described by:

$$\frac{df_{R_1}}{dt} = -C\left[\prod_i^N f_{R_i}\right]e^{-T_A/T}\ . \tag{18}$$

In this expression, $N$ is the number of reactants. The mass fraction of each reactant is $f_{R_i}$ for i = 1 to $N$, $C$ is the pre-exponential coefficient for the reaction, and $T_A$ is the effective activation temperature for the reaction. We see that the local rate of mass consumption of the primary reactant (per unit mass within a control volume) $df_{R_1}/dt$ increases with the local absolute temperature $T$ and the product of all the local reactant mass fractions. The primary reactant $R_1$, the number of reactants $N$, and the effective activation temperature $T_A$ for each reaction are given in Table 1. The effective activation temperatures for fuel cracking and soot combustion in Table 1 were taken from Said et al. (1997). The activation temperatures for fuel and intermediate species combustion were found by calibrating CAFE predictions by comparison to the data of Murphy and Shaddix.

**Table 1. Arrhenius Reaction Coefficients for a Typical Hydrocarbon Fuel Fire.**

| Reaction | $R_1$ | N | $T_A$<br>(K) | C<br>(1/s) |
|---|---|---|---|---|
| **Fuel Combustion** | F | 2 | 8,000 K | $10^{15}$ |
| **Fuel Cracking** | F | 1 | 42,730 K | $2 \times 10^{14}$ |
| **Soot Combustion** | C | 2 | 26,500 K | $10^{8}$ |
| **Intermediate Species Combustion** | IS | 2 | 8,000 K | $10^{15}$ |

Although the coefficients for the reaction stoichiometry and Arrhenius coefficients described above lead to good agreement with the data of Murphy and Shaddix, they should not be considered as the best possible set. Simpler single-step three specie combustion models may lead to similar agreement. The range of applicability of these coefficients is not known, except that they do lead to good agreement with experimental data in fires, which have temperatures in the range 800-1600K.

A second (optional) combustion model is available, and normally used along with the Arrhenius model described above. It is based upon an eddy dissipation (or breakup) concept (Magnussen and Hjertar, 1976). When using the eddy dissipation concept an eddy mixing time scale $t_{eddy}$ is added to the chemical kinetics time scale. The eddy mixing time scale is:

$$t_{eddy} = {C\,l^2}/{\varepsilon}\ , \qquad (19)$$

where C is a proportionality constant, *l* is the computational cell length scale (cube root of cell volume), and ε is the eddy diffusivity. A typical value of the constant C is 0.001, when using the CAFE-3D VLES model. This time scale is an expression of the time required to mix the contents of a computational cell by turbulent eddies. Note that when using the eddy breakup model the Arrhenius model is not turned off, instead the total time scale for the combustion reaction is the sum of the time scales for Arrhenius and turbulence. Since these combustion time scales act in series, the combination is less sensitive to errors in either model.

## 4.6 Radiation Transport Within and Near Fires

Thermal radiation transport is a mathematically intensive subject with many possible methods of solution. In CAFE, the objective is to obtain an approximate solution of suitable accuracy with a minimum amount of CPU overhead. To this end, in CAFE, thermal radiation transport within and near large hydrocarbon fires is split into two types: diffusive radiation inside the flame zone and clear air or view factor radiation outside the flame zone.

Thermal radiation transport within the flame zone can be categorized by the quantity of smoke produced by the fire. Clean burning fires, such as those fueled by alcohol or methane, could have long radiation absorption path lengths. Smoky fires, such as oil or jet fuel fires, have large amounts of soot, and consequently short radiation absorption path lengths. Recent measurements in a controlled JP8 fire indicated that the absorption length, $1/\beta_R$, where $\beta_R$ is the extinction coefficient and is on the order of 5-10 centimeter in a 2-meter fire (Murphy and Shaddix, 2003). Gases with long radiation absorption path lengths (compared to the relevant geometric scale length, i.e. fire diameter) are considered optically thin, whereas gases with short radiation absorption path length are considered optically thick – both types can be solved by different numerical methods. Optically thin media require numerical solution techniques that are typically CPU expensive, whereas optically thick media can use diffusion equation solution methods that are significantly less expensive. Media that are neither optically thin or thick require radiation transport solution method that are even more CPU intensive than optically thin methods. Within CAFE-3D only diffusive radiation transport algorithms are available. Thus, CAFE radiation models are only applicable to smoky fires – or those fires where the radiative absorption length is a small fraction of the fire diameter. Optically thin solution methods within the flame zone of a fire may be available in a future revision of CAFE-3D.

Outside the flame zone thermal radiation transport can be modeled by clear air or view factor methods. A complicating factor is that the surface of a fire, or smoky region, is dynamically changing due to the puffing and otherwise turbulent nature of flames. CAFE-3D includes view factor radiation model wherein the surface radiation characteristics of a smoky fire are recalculated each time step to allow clear air radiation transport. The model is simplified by ignoring shadowing effects of any intervening objects, and radiation absorption by clear air. Obviously, if a wall, or other object, separates a package from the fire, shadowing effects are very important. In such cases the user must explicitly account for shadowing effects, for example by turning off view factor radiation in shadowed locations. To further simplify and speed up the view factor computation, the surface normal vector of the flame surface is approximated to align with either the computational cell boundaries, or the vector sum of the surface normals of the cell boundaries if multiple surfaces of a cell are emitting radiation. Such approximations allow CAFE-3D to compute clear air view factor radiation without incurring a significant CPU overhead.

### 4.6.1 Diffusive Radiation Transport

There are currently two optional methods for including diffusive radiation transport within CAFE. Both methods utilize the Rosseland approximation for radiation transport in optically thick media. In the Rosseland approximation the thermal radiation transport is calculated by an effective gas thermal conductivity. The equation for the gas thermal conductivity is:

$$k_R = \frac{16 n^2 \sigma T^3}{3\beta_R}. \tag{20}$$

In this expression, $n$ is the medium index of refraction, T is the local temperature, $\sigma$ is the Stephan-Boltzman constant, and $\beta_R$ is the local extinction coefficient of the medium. The extinction coefficient is a function of the local mass fractions of soot, water vapor, fuel vapor, and intermediate species (Modest 1993). The first method is a simplified approach wherein the user inputs an effective constant Rosseland radiation absorption length. The second method calculates the effective radiation absorption length by including the absorption characteristics of soot, hydrocarbons, and temperature.

#### 4.7.1.1 Simplified Diffusive Radiation Transport Model

In the simplified Rosseland model, the user inputs a fixed value of the radiation absorption length ($1/\beta_R$). This fixed value is used at all locations within the fires. This simple diffusion model has the advantage of producing acceptable results for a wide range of problems without requiring the user to activate complex chemistry and species transport. A typical measured value of absorption length ($1/\beta_R$) for fuel oil fires is 0.1 meter (Murphy and Shaddix, 2003).

#### 4.7.1.2 Soot Dependent Diffusive Radiation Transport Model

The second diffusive radiation transport model includes most of the effects found in real fires. The effective Rosseland absorption length (1/extinction coefficient) is calculated by integrating the temperature dependent black body radiosity over the wavelength dependent soot absorptivity. As an option, this formulation also includes contributions for hydrocarbons, water vapor, and $CO_2$.

The formula for the Rosseland extinction coefficient of soot is:

$$\beta_R = \frac{3.6\ f_v\, C_o\, T}{C_2}\ , \tag{21}$$

where $\beta_R$ Rosseland extinction coefficient ($m^{-1}$) (1/L in conductivity formula),
$f_v$ is the soot volume fraction,
$C_2$ is 1.4388 cm K, (Plancks second constant), and
$C_o$ is a constant that depends on the soot complex index of refraction shown in the formula below:

$$C_o = \frac{36\pi nk}{(n^2 - k^2 + 2)^2 + 4n^2k^2}\,, \tag{22}$$

where n is the refractive index and
k is the absorptive index.

The soot volume fraction is calculated from the local soot mass fraction and the solid density of soot $\rho_s$ (approximately 1740 kg/m$^3$ Choi et al. 1995):

$$f_v = \frac{m_s \rho}{\rho_s}\,, \tag{23}$$

where

$f_v$ is the soot volume fraction,
$m_s$ is the mass fraction of soot,
$\rho$ is the local gas density, and
$\rho_s$ is the density (kg/m$^3$) of solid carbon in soot.

The mass fraction of soot is calculated by transport equation for soot.

### 4.6.2 Diffusive Radiation Boundary Conditions

Near a bounding surface, the Rosseland approximation is invalid. The use of jump boundary conditions has been applied by many investigators (Modest 1993, Seigel and Howell 1981). In CAFE-3D, jump boundary conditions are implemented by applying volumetric heat sources at all bounding surfaces equal to:

$$Q_v = \frac{\sigma(T_g{}^4 - T_w{}^4)A}{(3\beta_R\Delta/4 + 1/\varepsilon_w)V}, \tag{24}$$

where $\sigma$ is the Stephan Boltzman constant, $T_g$ is the gas temperature at the surface, $T_w$ is the solid surface temperature, $\beta_r$ is the local extinction coefficient, $\Delta$ is the distance between the cell centroid and the surface, $\varepsilon_w$ is the surface emissivity, A is the surface area of the material within or adjacent to the cell, and V is the volume of the cell.

This boundary heat source approach is nearly identical to the analytic solutions proposed by others. The difference is that the heat source is applied to the computational cell adjacent to the boundary rather than distributed amongst several cells. The error introduced by this approximation is small because a typical CAFE mesh uses computational boundary cells are larger that the mean free path of the thermal radiation. Thus most of the radiation exchange occurs in the boundary cell anyway. If mesh resolutions finer than the radiation mean free path are used, the result will be a slight temperature distribution error. However, the net radiant heat transfer would be almost unaffected. Further, the error introduced by this approximation is minor compared to the experimentally observed variations and temperature uncertainty in a real fire.

### 4.6.3 Clear Air View Factor Model

Outside the flame zone of a fire, heat transport is modeled by view factor methods. To do this, the fire is approximated as a dynamically changing solid object. This clear air thermal transport model is based upon the concept of establishing, and tracking the position of the surface of a smoky fire, and subsequently calculating the associated view factor radiation. The position of the surface is found by defining, and monitoring, a critical user input surface soot volume fraction. The surface soot volume fraction is a user input number, typically 0.5 PPM. The entire computational domain is examined for locations where the soot concentration between adjacent computational cells lies both above and below this critical value. Any cell boundary that satisfies the criterion is assumed to be part of the fire surface. The surface normal vector points in the direction of the lower concentration and is perpendicular to the face. In computational cells which have more than one radiating cell surface, a normalized vector sum is performed to assign the surface unit normal vector. When the problem is set up, all surface unit normal vectors of real solid objects are calculated and stored for later use. The fire surface unit normal vectors are calculated at each time step. The view factor between a fire surface cell and an object surface cell is calculated by the formula:

$$f_{12} = \frac{(\mathbf{n}_i \cdot \mathbf{s}_{ij})(\mathbf{n}_j \cdot \mathbf{s}_{ji})}{\pi \cdot s^4} A_j \, , \qquad (25)$$

where $\mathbf{n}_\mathrm{i}$ is the unit surface normal of the radiating fire surface,
$\mathbf{n}_\mathrm{j}$ is the unit surface normal of the object surface,
s is the distance between the surfaces,
$\mathbf{s}_\mathrm{ij}$ is the difference between the position vectors $\mathbf{r}_\mathrm{i}$ and $\mathbf{r}_\mathrm{j}$,
$\mathrm{A}_\mathrm{j}$ is the surface area of the object surface cell, and
$f_{12}$ is the view factor of the fire cell surface to and object cell surface.

This computation is rapidly performed using combinations of the dot product of the normal vectors and the position vector. Using the dot product eliminates the need for transcendental equation evaluations, thus the method is fast-running.

The proper choice of the critical surface soot volume fraction remains to be defined. Unfortunately, this parameter is neither fixed nor easily implemented as a function because it depends upon the computational cell size, (that is if large cells are used at the perimeter of the fire). Measurements of the emissive power of open pool fires, as a function of pool diameter, have been made by many people in the fire community. A plot of average pool fire surface temperature, derived from the measured emissive power (SFPE 1995), as a function of pool diameters is shown in Figure 2.

For a given fire size (and computational mesh structure), the user can adjust surface soot volume fraction until the predicted fire surface temperature and the measured value are the same. The dependence of fire surface temperature upon computational cell size is due to the fact that larger cells will grab a greater amount of air, thus diluting the radiant energy and resulting in lower surface temperatures. Using small cell sizes will avoid this complication, but this may result in an unacceptably slow computation. Thus, depending upon the problem being solved, the user may be required to run several calculations varying the surface soot concentration until a suitable match between predicted and measured surface temperatures is found. Typically, the critical surface soot volume fraction is in the range 0.5 PPM and has been found to be fairly constant from one fire to the next when compared to experiment. The greatest sensitivity appears to be mesh resolution, larger fires requiring slightly lower values.

If the user is modeling engulfed objects, the dependence of engulfed heat transfer to the surface soot volume fraction is a weak one. The weak dependence is caused by a negative feedback effect. If the user selects too high a value, the fire size will be smaller but hotter. Conversely if

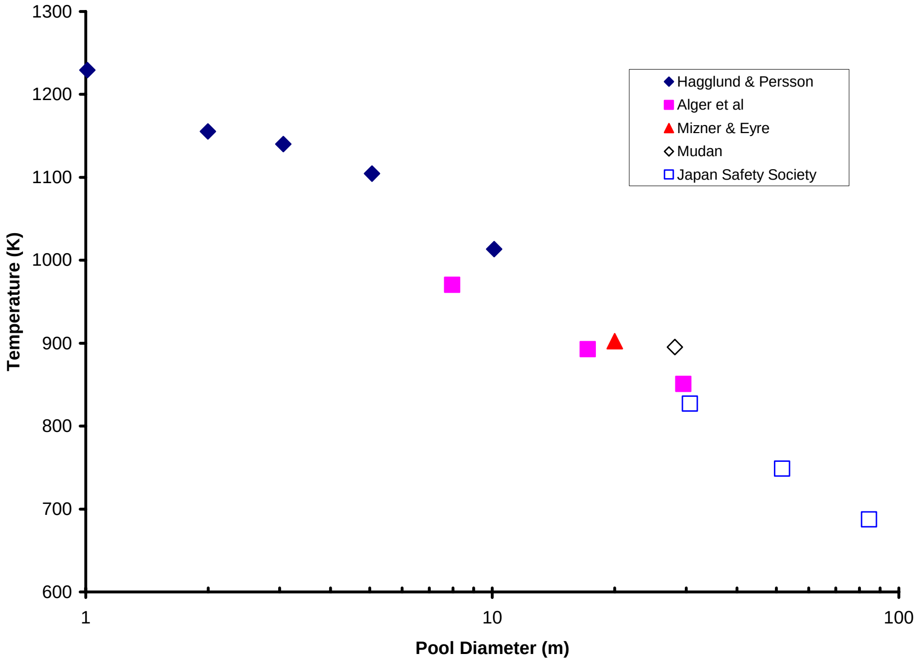

**Figure 2. The heat flux weighted average surface temperature of an open pool fire**

the user selects too low a value, the fire will be larger and cooler. In both cases the total emitted radiation would be similar, resulting in a similar energy balance (and therefore temperature) for the fire resulting in a lower sensitivity to radiant heat transport inside the flame zone.

## 4.7 One-Dimensional Subgrid Models

The CAFE porosity model assigns polygon surfaces to solid objects that intersect the computational cells as previously shown in Figure 1. CAFE includes a set of one-dimensional CFD solvers that can be assigned to the set of polygon surfaces. Thus, local grid refinement for solid surfaces is modeled in one dimension and it can exchange mass, momentum, energy, and species with the three-dimensional mesh. In this way, various heat and mass exchange processes can be simulated on a small scale compared to the computational grid size. There is no limit on the number of 1-D subgrids (provided one is able to re-compile), nor how many 1-D cells can be assigned to each polygonal surface. The primary limitation is that the one-dimensional subgrids do not communicate with their 1-D neighbors. The heat and mass transport is directed normal to the polygon surface, not laterally. There is no restriction on the number of different types of subgrids. Some can model a steel surface, others composite layered materials, and others chemically reactive materials such as wood. All can be active in the same problem.

As an example, if the user wanted to burn wood, 1-D subgrids would be assigned to the 3-D model of the wooden object. 1-D conservation equations (mass, momentum, energy and species) would be activated, and mass/energy/species exchange coupling assigned to the 3-D mesh. As the wood is heated, say for example by a pre-existing fire or other heat source, chemical reactions would decompose the wood into char and combustible gases. The combustible gases would be transported to the boundary of the 1-D mesh and then passed into the 3-D mesh where they can react with oxygen to form a wood-burning fire. Since the 1-D mesh includes species equations, depletion of the combustible gases during the pyrolysis reactions would be tracked and the wood combustion process would slowly diminish and ultimately decay into glowing char. Of course, the user would be required to set up all of the chemical reactions associated with such phenomena. The point is that such a framework exists within the CAFE code.

In a typical CAFE calculation the user is more interested in heat transfer to solid objects. In such a case the 1-D subgrid momentum and species equations are turned off and only the 1-D subgrid energy equations are activated. When CAFE is coupled to P/THERMAL, only the first defined group of 1-D subgrids will be coupled to the P/THERMAL model. The other subgrids can be used for other purposes, or not used at all.

In the example problem (see Section 11), 1-D subgrids are used to track the thermal heat-up of a thin (2 millimeter) steel enclosure and of a 1 centimeter thick calorimeter. Shown are the surface temperatures after a two minute fire. These temperatures were calculated by heat exchange between the 3-D grid, which includes the fire and its associated thermal radiation, and the 1-D subgrids, which includes the transient heating in the comparatively thin objects.

## 4.8 CAFE-3D Code Validation

The CAFE code has been tested both by comparison to analytical and experimental data. Theoretical comparisons of conservation of mass, energy, and species have been performed with good results. Overall (global) conservation of mass is accurate to machine accuracy. Local conservation of mass is accurate to the user-specified EOS error (an input parameter). Local and global conservation of energy and species depend upon the problem being solved. Steady flow (combustion) problems have energy and species conservation accurate to within a percent. Transient combustion problems have energy and species conservation accurate to within a few percent, or better depending upon the problem. The loss of accuracy for transient problems is due to the segregated nature of the solver, that is each conservation equation (momentum, energy, species) is solved in turn, as sequence of steps, rather than simultaneously. Note that these comparisons are done over long time scales where errors can accumulate. Conservation errors over a single time step are negligible.

Conservation of momentum (which include all the terms in the momentum equation simultaneously) cannot be tested directly by numerical computation because solutions to the Navier-Stokes equations are not unique. There are no theoretical solutions except for trivial cases such as laminar flow. Instead, conservation of momentum has been validated by comparison to experimental data. Flow through orifices and around blunt bodies was examined and comparisons of pressure distributions and vortex shedding frequencies were made. Results were within experimental error.

Global/integral comparisons which include all of the conservation equations, the 1-D subgrid equations, as well as sub models such as heat transfer, radiation, combustion, etc. have been made by comparisons with several open pool and enclosure fires. Results of these comparisons were good and are discussed in Greiner and Suo-Anttila (2004), Lopez et al. (2004), Suo-Anttila et al. (2004), Ju et al. (2003), Suo-Anttila et al. (2001), and Lopez et al. (2003).

Note that some of the papers refer to Isis or Isis-3D, which as stated earlier is identical to standalone CAFE-3D.

Most of these papers show comparisons of heat transfer measurements within and near fires with code predictions. The Suo-Anttila et al. (ITSC-2001) report contains numerous examples for a wide variety of theoretical and experimental comparisons, including the conservation equation comparisons.

This page intentionally left blank.

# 5. Using the CAFE-3D Code

In order to use CAFE-3D, the user must first build a P/THERMAL model of the object. The CAFE-2D user manual (Suo-Anttila 2000) includes an appendix that describes how to build a P/THERMAL model of a calorimeter, step by step. When assigning boundary conditions to the object the user should select all the external surfaces and assign them to a template-defined convection boundary condition.

In an auxiliary file, template.dat, a user ID (UID) convection configuration number must be assigned, and that number must be greater than or equal to 1001. Assigning a number of 1001 or greater tells P/THERMAL to call the ulib.a subroutines for convective heat transfer boundary conditions. P/THERMAL often creates many different template.dat files. The user must check the working directory where CAFE-P/THERMAL is running for the appropriate template.dat file.

At the present time CAFE does not have an automatic mesh generation option to build an equivalent model from the P/THERMAL model. Instead the user must build a CAFE model that has approximately the same external size, shape, and physical orientation. CAFE has a solid modeler, which allows the user to construct a complex object by adding and subtracting a few graphics primitives. Internal details of the solid object are unnecessary because they are accounted for by the P/THERMAL model.

Within the CAFE input file, there is an option to automatically assign a 1-D subgrid to all structural surfaces. This option must be used by CAFE-3D in order to couple the finite element faces on the surface of the PATRAN object to the equivalent surfaces in the CAFE-3D model. Additional details of how this is done will be described later in this document.

Once the user has built an equivalent CAFE model, CAFE standalone is run for a short period of physical time, typically 10-20 seconds, at the end of which a restart file is written. This initial CAFE run allows the object to become engulfed in flames, which form a suitable starting point for the CAFE-3D run.

When running CAFE, there must be four files in the working directory, ulib.a, template.dat, I3d.inp, and restart.inp. CAFE can be run from within PATRAN or from a command prompt using the patq utility. In each case the user must be aware of, and work with, any subdirectories that PATRAN creates and uses as a working directory.

The thermal results of the CAFE-3D run can be viewed by PATRAN. If GMV or Tecplot are available, then three-dimensional fire isosurfaces can be viewed for the duration of the CAFE run provided that output option was activated. In addition to GMV and Tecplot, Xplot can be used to view animated contour (2D) slices through the fire at pre-selected locations. See Section 6.10.2 and 6.10.3 for a description of GMV and Tecplot, respectively. The Xplot software can be obtained by contacting the authors of this manual.

## 5.1 Mapping PATRAN Boundary Conditions to CAFE-3D

The boundary conditions that are assigned in PATRAN must be mapped to the equivalent object surface boundary conditions in the CAFE-3D model. The coordinate systems in PATRAN and CAFE are co-located, which means that the X, Y, and Z axes are aligned in the same direction with respect to the object. Thus when the user is building the model the orientation of the object with respect to the X, Y, and Z axes must be the same. It is best to orient the object correctly when first building the object in PATRAN. However if the object is already built, or the user forgets to orient the object, then it may be rotated into the correct orientation with either PATRAN or CAFE. If the user chooses to rotate the CAFE object, then a new restart file must be created.

The mapping of the P/THERMAL model created in PATRAN to the CAFE model is done automatically. The algorithm employed by CAFE-3D is to non-dimensionalize both the P/THERMAL and the CAFE-3D model in the X, Y, and Z directions. Using the non-dimensional positions, all of the FE surface nodes are mapped to CAFE model surface nodes. The closest non-dimensional position in the X, Y, and Z directions is chosen for the association. Note that if the P/THERMAL and CAFE-3D models are not co-located with respect to the X, Y, and Z axes, the mapping will still occur, however the result will be a very peculiar temperature distribution on the surface of the object.

The automatic mapping may result in some CAFE nodes that are unassigned depending upon the grid resolution of the corresponding P/THERMAL and CAFE models. Unassigned CAFE nodes will have an adiabatic boundary condition with respect to the fire. If too many unassigned CAFE nodes result, the user should consider either refining the PATRAN mesh, or coarsening the CAFE mesh. The number of unassigned CAFE nodes appears in the CAFE output file (I3d.out).

The P/THERMAL to CAFE mapping can be viewed graphically with the Geomview code. Geomview is an interactive 3D viewing program that can be obtained for free from http://www.geomview.org. CAFE will output a file called Map.skel, and this file can be rendered graphically by Geomview. All of the PATRAN nodes are displayed and color-coded. PATRAN nodes that share the same CAFE node will have the same color, thus the user can graphically see how well the models are mapped to each other.

# 6. CAFE-3D Input File Description

The input for CAFE-3d consists of an ASCII text file called I3d.inp, or optionally I3d*.inp, where * represents an ASCII character string. The maximum length of the filename I3d*.inp is 40 characters. The input file can be edited with any text editor. There may be additional input files that have time dependent information (Table.inp; see Chapter 6.11), spatial distributions, or radiation view factor data. If additional input files are required, the format will be described in the appropriate section.

Typically a user does not build an I3d.inp file from scratch, rather the user would pick an existing file that has similar features to the problem at hand. The I3d.inp file would be modified to suit the new problem. This saves considerable debugging time, since the user would presumably have a file that already works.

Comments can be inserted anywhere in the I3d.inp file by placing a # sign or the letter c in column 1. Comment lines can be placed at any location in the file. This allows the user to place as many reminders of correct format structure as required to write input files correctly without referring to this manual.  Note that the input file must not contain empty lines or there will be reading errors.  Instead, make use of “#” or “c” at the beginning of the line if space between input lines is desired.

## 6.1 Control Keywords

A * in column 1 indicates a control keyword.  I3d will attempt to read the control keyword and that will determine the type of input, which follows. Keywords are not case sensitive, but the spelling must be correct. In addition, the order in which control cards appear is important. The correct order is:

*Time
*Equations
*Regions
*Object3d
*Object1d
*Coupling
*Output

After each control keyword there may be additional sub-section control keywords. Some of these subsection control keywords are optional. Skipping an optional keyword causes default values for those parameters to be used in the code.

## 6.2 Typical CAFE Input File Example

The input file structure for CAFE is shown below. Normally CAFE input files are not this large. This example was generated to include most of the possible input variations so that a user can quickly see the correct input format. This example is not internally consistent, that is pointers may be defined to parameters or other objects that may not exist. In fact it cannot be made internally consistent because there are too many options, some of which will conflict with each other. The intent of this example is to demonstrate the proper format for input. It is not intended as an example that could be executed successfully by the code. A simpler working example that can be copied and pasted into a text editor (provided you have an electronic version of this document) is presented in Section 11 of this manual.

```
#
#  Comment section of the input file
#  any line that begins with  #  or c  is interpreted as a user comment.
#
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
*debug  ‘z’
*Time   nmax    ipstep  time    restart
       100000     5       0       'w'
c       #       tstop   dtmax   dtmin         courant dtemp   drhog   dxmu    dtwall
                 1      0.01    1.00E-04        0.5     1.2     1.2     1.5     1.2
                 10     0.03    1.00E-04        0.5     1.2     1.2     1.5     1.2
                 100    0.3     1.00E-04        0.5     1.2     1.2     1.5     1.2
*end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
*Equations
c
*Momentum  Type    uwm     xwm     ncor    errmx           sor     Gx      Gy      Gz
       'pressure'  0       0.1     12      1.00E-03        1.7     0      -9.8     0
*Momentum  Type     eps1d    courant1d
           '1d'       .5        100
*Energy type    uwe     iter    sor
       '2eqn'   0.4     10      1.7
*Species  name         uws   iter   sor    W  1-d    1-d    conserve
      6   'fuel'        1     10    1.5   50  'on'  'off'   ‘off’
          'o2'         .5     10    1.5   32  'on'  'off'   ‘off’
          'soot'       .5     10    1.5   29  'on'  'off'   ‘off’
          'intr'       .5     10    1.5   29  'on'  'off'   ‘off’
          'h2o'        .5     10    1.5   18  'on'  'off'   ‘off’
          'co2'        .5     10    1.5   44  'on'  'on'    ‘off’
*Chemistry nreac  type       #Sp coef  spec   coef      name
             3  'Sootmodel'   2  1.0   'fuel'  2.87     'o2'
                              3  0.125 'soot'  3.82     'oil_pc'  .2 'intr'
                                 1e15   8000   37.7e6  300
                              1  1.0   'fuel'
                              2  0.05   'soot'  .95      'intr'
                                 2e14  42730   -3e5
                'Combustion'  2  1.0   'intr'  3.07     'o2'
                              1  4.07  'oil_pc'
                                 1e15  8000     42.12e6   310
                'Combustion'  2  1.0   'soot'    2.6    'o2'
                              1  3.6   'oil_pc'
                                 1e8   26500  32e6      400
#       Fv_min   esoot   tinf   'view'  nvapors
*Soot    .5e-6     1     477     'on'      4     'fuel'  'o2'  'soot'
 'fuel' 'CH4'  5.1  'intr' 'CH4'  5  'soot' 'SOOT'  5.55 'oil_pc'  'CH2_PC' 1
*Diff_reac
  .004
c
*End
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
*Regions
```

```
c
*Fluid  Type    w       gam     pres0           fluid name
       'igas'  29      1.4     1.00E+05            ‘air’
c-----------------   fluid thermophysical properties     ----------
*GasProp
            'Cp'    'spec_temp' 'air' 3  300   1000
                                         1580  1250
                                         2000  1350
                               'fuel' 2  300   2200
                                         2000  2200
                                'o2'  3  300   1000
                                         1580  1250
                                         2000  1350
                               'soot' 2  300   2000
                                         2000  2000
                               'intr' 2  300   2200
                                         2000  2200
                             'oil_pc' 2  300   1100
                                         2000  1500

     'Visc'    'data'    4   250   2e-5
                             500   2.5e-5
                             1000  3e-5
                             2000  4e-5
       'Cond'  'Radiation'     1       0.05
       'Cond'  'PwrLaw'        4.81E-04        0.7
c------------------   fluid initial conditions    ---------------
*GasReg #       press      temp    rux     rvy     ruz     m1      m2      m3      m4
       1       1.00E+05    300     1       0       0       0       0.23    1.00E-08 0
       2       1.00E+05    300     0       0       0       0       0       0        0
       3       1.00E+05    300     0       0       0       1       0       0        0
c------------------  structure material properties  --------------
*Struc Point Key temp   Kg/m3    K     c  M2/m3  emis h tinf alpha  name  mass-frac  name
c              ---temperature and species dependent need entries for each species---
       1  'VAR'   295.  1865.              0.     0   0  0     1   'IHE' 0 0 0 0 0 0
                    'tk'  'Spec_Temp' 2  0   .75
                                         681  0.0
                                      3  100 .16
                                         300 .16
                                        4000  1.5
                                      1  300   0
                                      2  300 .16
                                        1300 1.4
                                      1  300   0
                                      2  100 .18
                                        2300 .19
                    'cp'  'Spec_Temp' 2  0    236
                                         1000 2936
                                      2  0    236
                                         1000 3936
                                      2  0    721
                                         1000 1673
                                      2  0    336
                                         2000 5936
                                      2  0    321
                                         1500 6673
                                      2  0    236
                                         1000 2936
c                ---temperature dependent---
           3  'Var'   300.  1272.                0     0   0   0     1  'fuel'   0 0 0 0 0
                     'tk'   'data'   4   100  .1
                                         500  .6
                                         1200 1.3
                                         3000 3.5
                     'cp'   'data'   3   100   800
                                         1000  1200
                                         3500  1900
```

```
c       Key   temp   Kg/m3    K     c  M2/m3    h    emis tinf alpha
    2  'Con'   295.  1890.    15 1586     0.    0     0.  0.    1  'insulator1'  0 0 0 0 0
    4  'Con'   300.  803.     0   1e4     0     0     0   0     1  'insulator2'  0 0 0 0 0
    5  'Con'   300.  800.    .03  40.     0     0     0   0     1  'ceramic'     0 0 0 0 0
    6  'Con'   300   2700   200  1e10  100     0     0   0     0  'heat sink'   0 0 0 0 0
c------------  Heat and Mass Sources   ----------------------------------
*Power Pointer  Key     power    period    power-on pwr-off  fgw
         1     'Exp'   1.00E+05   0.001      0        100      1
c                       power     fwhm     center            fgw
         2   'Gaussian' 6.52e10   .005     0.025              1
c table.inp heat source W/m^3  scale     data-column Fraction-in-gas(fgw)
         3    'Table'          1e8          2              1
c table.inp-mass-source Kg/sec   scale    data-column    Temp-source
         4    'mass'             37.8         3              450
c---------------  Heat transfer correlations  ---------------------------
*Heat Pointer  Key
c           constant h
       1       'H'            5
c       laminar-turbulent  A  *Ren**B *Pr**c + d   hdia
       2       'lam/turb'  64      -1    .33  0.0   1.3
                2800      .023     .8    .33  0.0
       3       'Reynolds' .05      .6    .4   2.0   0.3
c               Type      Rho*Vel   Hdia
       4       'Stag'        0      1.3
       5       'Cyl_Cross'   0      1.3
       6       'Wall'        0       6
c  Nusselt-Grashoff        delta-T    Hdia
        7       'Grash_v'      0      5
        8       ‘Grash_h’      0      2
        9       ‘pipeflow’            1
c---------------  Drag correlations  ------------------------
*Drag Pointer  Key       a  *Ren **B + c     Hdia
        1   'Reynolds'  1.0       -1  0.0   2.6e-5
        2   'Lam/Turb'  1.0       -1  0.0   1.4
              2800      .064     -.5  .03
        3   'DragCoef'  2.4
        4   ‘Moody’    .0001   2
c---------------  Turbulence Models  ------------------------
*Turb Reg  Key    Type  coef len scale=(ax (x - x0) + by (y-y0) + cz ( z-z0) ) **d  +  e
      1  'Eddy'  'LES2'   .05           0      0     .1     2.8   0      0      1     .001
      2  'Eddy'  '1-Eqn'  .6   .164     0      0     .1     2.8   0      0      1     .001
      3  'Eddy'  'Ren'    .1    .5      0     1.2
      4  'Eddy'  'LES'    .2
-c---------------  Orifice Definitions  ---------------------
*Orfice Pointer  Key     %Areax  Coefx   %Areay  Coefy   %Areaz  Coefz
           1   'Normal'    0       0       0      1      0     0
           2   'Normal'    0       1       0      0      0     0
c time dependent     Column-pointer
           3   'Time'      1       2       0      0      0     0
*endreg
c------------------End of modeling Parameters-------------------
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c------------------3D Mesh Creation Parameters-------------------
*object3d  'Fire'
*mesh
      'nxnynz'   0 0  0
# x grid
      '-xpwr'   0      8.7554      12
       'xfix' 8.7554   8.9855       1
       'xfix' 8.9855  11.0145       6
       'xfix' 11.0145 11.2446       1
      '+xpwr' 11.2446    20        12
# z grid
      '-zpwr'   0      8.7554      12
       'zfix' 8.7554   8.9855       1
       'zfix' 8.9855  11.0145       6
       'zfix' 11.0145 11.2446       1
      '+zpwr' 11.2446    20        12
# y grid
       'yfix'     0    .3365        2
       'yvar'  .3365   4.4005       8
       'yfix' 4.4005   4.88315      1
       'yfix' 4.88315  5.11175      1
       'yfix' 5.11175  5.4516       2
       '+ypwr' 5.4516   10          8
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c------------------  3-D model definition--------------------------------
c
*geom  type  xmin xmax  ymin ymax  zmin zmax             Gas Wall  Pwr  HT Turb  Drag  Orf
 'domain'      0   20     0   10    0   20                1    0    0    1   1    0     0
    'box'      5    6     3    9    0    2                1    1    2    1   1    0     0
c Ellip-Sph axis  x0   y0    z0  ymin   ymax  rady   radxz
'open_esph  'y' 10 5.11175 10 5.1175 5.4516 .3341 1.0145  1    2    0    1   1    0     0
c Cylinder  axis x0   z0   ymin     ymax     rad
'open_cyl'  'y'  10   10  4.88315  5.11175   1.0145       1    2    0    1   1    0     0
     'cyl'  'y'  10   10  .3365    4.88315   1.2446       1    2    0    1   1    0     0
     'cyl'  'y'  10   10  .0       .3365     1.0145       1    2    0    1   1    0     0
c   Sphere   x0   y0   z0    rad
     'cyl'   .1   3    4.4   .56                          1    2    0    1   1    0     0

c ellip-cyl axis x0   z0   ymin     ymax     radx   radz
'ecyl'      'y'  10   10  .3365    4.88315   1.2     3.3  1    2    0    1   1    0     0
c Plane xmin xmax  ymin ymax  zmin zmax (=clip box)      Gas Wall  Pwr  HT Turb  Drag  Orf
'Plane'   2    4   1.4   6.3    1   10                    1    2    0    1   0    0     0
c x1 y1 z1    x2  y2  z2    x3  y3  z3---  3 points define a plane
   8  1  1     1   9   2     1   1   7
c 3 points on the plane
   1  2  3      3  3  3      8  5   2
c  Sphere  sphere-center  radius
'Sph'       5    5   5     3.4                            1    2    0    1   0    0     0
c
c------------------  3-D model boundary conditions  -------------
c
*Boundaries     Side         Type   start stop    start stop   Arg1   Arg2   Arg3 ......
               'a'          'flux'    0      0      0     0     0      300
               'a'          'conc'    0      0      0     0     0      0.23    0   0   0
               'left'       'flux'    0     10      0    20     1      300
               'right'      'hydro'   0     10      0    20    300
               'top'        'hydro'   0     20      0    20    300
c Radial  pool  xc   yc  rad  tboil, Hfg,    Emiss,  Rate0 Time0 Max  Min Rate
'rbot'  'evap'  10  10  3.58  450   3.00E+05  0.5    0.072   2   0.25 0.01
'rbot'  'conc'  10      10      3.58    1  0  0  0  0
*end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c------------------  1-D user defined subgrid model -------------------
*object1d    'Nozzel'
*1Duser  nxi  nxf   ji   jf       Gas Wall  Pwr  HT Turb Drag Orf   Len
          1    10    1   20        1    1    0    0   0   0   0     .2
*Boundaries  Side     Type   Arg1  Arg2  Arg3
            'left'   'press3d'   0      0
            'right'  'flux'     10    100
*end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c--------------  second 1-D user defined subgrid model ------------------
c
c  *1DSuser is a sphere  inner radius to left(x=0), outer to right
c  *1DRuser is a cylinder
c  *1Duser  is a cartesian coord 1-D object
c
*object1d    'Radial Nozzel'
*1DRuser  nxi  nxf    ni  nf      Gas Wall  Pwr  HT Turb Drag Orf   Rinner  Rlen   Frac
          1     50    21  31       1    4    0    0   0   1   0       0    .00635   1
*Boundaries  Side     Type   Arg1  Arg2  Arg3
```

```
            'left'   'press3d'   0      0
            'right'  'flux'     10    100
*end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c--------------------- automatic 1-D subgrid model -------------------------
c
*object1d    'Cal'
*1Dauto  nxi  nxf                 Gas Wall  Pwr  HT Turb Drag Orf   Len
          1    10                  1    1    0    0   0    0   0     .1
*Boundaries  Side     Type   Arg1  Arg2  Arg3
            'left'   'flux'   0      0
            'right'  'flux'   0      0
*end
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c--------------User Defined Coupling between 3-D and 1-D subgrids --------
*Coupling  'user'                     position        dir cosines
#
# CO2 Nozzles
#
#Surf typexfr1 name        region     x    y   z      xc    yc     zc   emiss
  1   'qms'    'Fire'      'point'  74.0 21.0 1.32   0.0   0.0   -1.000 1.0
#                Arg1       Arg2     Arg3       Arg4
               'Nozzel'   '1du'    'right'       1
  2   'qms'    'Fire'      'point'  70.3 21.0 1.32   0.0   0.0   -1.000 1.0
               'Nozzel'   '1du'    'right'       2
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c---------------Automatic Coupling between 3-D and 1-D subgrids-----------
*Coupling  'auto'
#surf typexfr1 name        region  face1   xi  xf  yi yf   zi zf emiss
  1   'q'     'Fire'      'box'   'g-s2'    0  30   0 15   0  30   1
              'Cal'       '1da'   'left'    1                       1
*end
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*object1d    'Pool'
*1Dauto  nxi  nxf                 Gas Wall  Pwr  HT Turb Drag Orf   Len
          1    10                  1    2    0    0   0    0   0    .1    'pool'
*Boundaries  Side     Type   Arg1  Arg2  Arg3
c                             T   Hfg  fuel_Cp Ev_0   t0  Ev_mx Ev_mn   mass frac
            'left'   'evap'  450  3e5  2200    .01    2e6  .25  .002   1 0 0 0 0 0
            'right'  'flux'   0      0
*end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c----------------Output Parameters-------------------------------------------------------
*Output
c---------------Tecplot-----------------------------------------------------------------
*TEC3D delay   dt  xi  xf  yi  yf   zi  zf
         0     .1   0  35   0  27    0  32
c---------------GMV---------------------------------------------------------------------
*GMV   delay   dt  xi  xf   yi  yf   zi  zf
         .0   .05   0  60   0   25    0  60
c-------------ACSII Formatted USER Defined subgrids--------------------------------------
*Ascii_1du     delay   dtplot ndat    i      j
               0.0      .1    2      3      8
                                     1      6
c-------------ASCII Formatted at x,y,z positions-----------------------------------------
*Ascii      pdelay dtplot  ngroup
             0.0    .05      1
            'PoolCenter'    5   x     y     z
                                0   3.04    0
                                0   3.29    0
                                0   4.29    0
                                0   4.79    0
                                0   6.79    0
c-------------ASCII Formatted AUTO 1-D subgrids at specified x,y,z coupling locations-----
```

```
*Ascii_1d   pdelay dtplot  ngroup
             0.0    .05     3
               'WallHFG'    4   x     y      z   1d
                              -2.75  2.85    0    5
                              -2.75  3.85    0    5
                              -2.75  4.85    0    5
                              -2.75  5.85    0    5
               'PoolHFG'    4
                              1.015  2.85    0    1
                              1.015  3.85    0    1
                              1.015  4.85    0    1
                              1.015  5.85    0    1
               'Calorim'    4
                               .00   3.94    0    1
                              -.16   4.10    0    1
                               .00   4.26    0    1
                               .16   4.10    0    1
c-------------Binary XPlot 1-D user subgrids distribution--------------------------------
*1DU        delay    dtdist  #    start  nstop  1dU
             0.0      .2     3     1     10     1
                                   1     10     12
                                   1     10     15
c--------------Binary XPlot Auto 1-D subgrid distribution at specified x-y-z location-----
*1D       delay    dtdist   #     x      y     z   start stop
           0.0       1      4   39.45  .61    30    1    20
                                39.45  1.83   30    1    20
                                39.45  3.00   30    1    20
                                39.45  4.00   30    1    20
c------Angular Surface distribution  Temperature and Heat Flux --------------------------
c*Surf1d  = Binary XPlot formatted
c*Surf1d_ASCIII  = ASCII formatted
*Surf1d     delay   dt   number   axis    xmin   xmax    ymin  ymax
              0     .5     1       'z'     0       6      2      4
c--------------Binary XPlot line distribution in 3-D grid parallel to x, y, or z axis-----
*Binary-1D # delay  dtdist   #   start  nstop   I2  I3  dir
               0       .1    2      2      6    26  15  'y'  ‘flame zone’
                                    2      6    26  16  'y'  ‘edge zone’
c--------------Binary XPlot 2-d contour slices or projections of 3-D grid-----------------
*Binary-2D   #   delay  dt
             4   0.0   .5
#Binary 2D  filename      axis   start  stop  start   stop  plane
             'pl2dz.bin'    'z'     0     30    0      30    15   ‘specz.bin’
             'pl2dy.bin'    'y'     0     30    0      30     0   ‘specy.bin’
             'pl2dy2.bin'   'y'     0     30    0      30     3   ‘specy2.bin
             'pl2dx.bin'    'x'     0     30    0      30    15   ‘specx.bin’
*end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c------------END of Input File-----------------------------------------------------------
```

The various control regions in the input file have been separated by a line of comments ‘cccccccc,’ in order to make the appearance of the file more readable.

In the following format description, an appropriate section of the input file will appear as a short example in the description section.  Thus, the user need not flip back and forth among the pages of this document. In addition, each section, subsection, and data line will be highlighted. Optional items will be enclosed in square brackets [...].

## 6.3 Time Step Control

The first section of the input file defines the time step control parameters for the problem. An example of this section of the input file is shown below.

```
###########################################################################
*Time  nmax     ipstep  time      restart
      111000       5       0        'r'
#  #  tstop  dtmax  dtmin courant dtemp drhog  dxmu  dtwall
        5.01  .01    1.e-3   .5    1.2   1.2   1.5   1.2
        100   .1     1.e-3    1    1.2   1.2   1.5   1.5
*end
###########################################################################
```

The time step control parameters are defined below.

<u>*Time  nmax  ipstep  time  restart</u>

*Time

A control card telling the code that time step control parameters follow.

nmax

The maximum number of time steps. The problem will terminate when nmax is reached or when the physical time equals tstop, whichever occurs sooner.

ipstep

The number of time steps between screen prints.

time

The initial start time.

restart

A problem restart parameter. This is a character input, which is read in free format so it must be surrounded by single quotes. Valid values are:

'n' None. Do not read or write a restart file,
'r' Read a restart file,
'rt' Read a restart file, and reset problem start time to zero,
'rw' Read a restart file and write one at the end of the calculation,
'rwt' Read a restart file, reset start time to zero, write restart at end of calculation,
'w' Write a restart file at the end of the calculation,
'rft' Read fluid flow only, and reset start time to zero, or
'rst' Read structure temperature only, and reset start time to zero.

When choosing to restart a problem the user can change some parameters in the input file but not all of them. For example, the boundary conditions can be changed, run control parameters, and modeling parameters (such as heat transfer coefficients, material properties etc.). Things which may **not** be changed in a restart are those which would leave regions undefined such as a change in the grid structure, number of computational cells, the location, size, shape, or number of solid objects,  number of chemical species, and number of 1-D subgrids. Note that subsequent restarts of CAFE overwrite output files that were written during earlier runs.

The next line(s) of data contains time step control data. The user can enter any number of lines of time step control. The *end keyword tells the code when time step control data is complete. Some problems may be best run with different values of time step control, for two reasons; (1) to control the time step for problem stability, and (2) to obtain intermediate restart files.

For problem stability, the user can run a problem with a small Courant number or other time step control until rapidly varying dynamic phenomena stabilize, and then switch to another value at later times.

If the user wants to create restart files at multiple times, the code will produce a restart file at the end of each time zone. The file name is called restart*.inp, where * is a number corresponding to the time zone. The maximum number of restart files is 99, and the last restart file is called restart.inp.

In the example shown above, three lines of time step control are included.

tstop dtmax dtmin courant dtemp drhog dxmu dtwall

Tstop

The stop time of the calculation, or if multiple time step control lines are read, then the time over which the current time step parameters apply. The problem will terminate at the final tstop time.

Dtmax

The maximum allowed time step.

dtmin

The minimum time step during the calculation. This parameter is also the problem start time step size. If time step controllers, such as the Courant condition, require a smaller time step than allowed by dtmin then the problem is terminated and a message is printed.

Courant

The Courant-Friedrich-Lewy or CFL number. The fraction of a cell size that can be traversed by the velocity of the fluid in a single time step. This number can exceed 1 but may lead to poor transient accuracy, or the solution may become unstable, under some conditions.

Dtemp

The maximum fractional change in fluid temperature $(T^{n+1} - T^{n}) / T^{n+1}$ during a time step, where n= time step number. Recommended value is 1.2.

Drhog

The maximum fluid density change per time step. Recommended value is 1.2.

Dxmu

The momentum viscosity parameter time step control. Recommended value is 1.2.

Dtwall

The maximum structure temperature change per time step. Recommended value is 1.2.

*end

The end card for time step control parameters.

## 6.4 Governing Equations and Models

This section of the input file tells the code which equations to solve and which models to activate. This includes momentum, energy, species, chemical reaction, 1-D momentum and a soot model. The user does not need to activate all of the equations. It is recommended that only the equations that need to be solved are activated in order to keep the code running quickly.

```
##########################################################################################
#
#
*Equations
#
*Momentum    Type     uwm  xwm  ncor  errmx   sor   Gx      Gy  Gz
        'pressure'      0  .05   10   2.e-4   1.5    0   -9.8   0
*Momentum  Type     eps1d    courant1d
           '1d'       .5        100
*Energy   type         uwe   iter   sor
         '2eqn'        .3      5    1.5
*Species  name         uws   iter   sor    W    3d    1d    ‘conserve’
      2   'fuel'        1      5    1.5   50   ‘on’  ‘off’    ‘off’
          'o2'         .5      5    1.5   32   ‘on’  ‘off’    ‘off’
*Chemistry nreac  type       #Sp coef  spec   coef  name
             1  'Combustion'  2  1.0   'fuel'  3.0   'o2'
                              1  4.0   'null'
c                               Pre-Exp -B/R     Hc   Tmin
                                 9e12    8000   30e6  250
*Endeqn

#
#############################################################################################
```

### 6.4.1 Governing Equations

*Equations

This control card tells the code which equations and models to activate. There are five choices: momentum, energy, species, chemistry, and soot. Only those choices that appear in this control section of the input file will be solved when running the problem.

*Momentum Type uwm xwm ncorr errmx sor Gx Gy Gz

*Momentum

Type

The type of momentum solution. There are three options: ‘Constant,’ ‘Pressure,’ and ‘1-D.’

‘Constant’ includes no-flow solutions where all velocities are zero, such as in a conduction solution. Constant also includes velocity distributions imposed by the user, such as a constant velocity everywhere. Constant can also be used when reading a velocity distribution from a restart file that was generated by a previous run and the user wants the velocity distribution to remain fixed for the entire calculation.

A 'Pressure' option tells the code to solve for the momentum distribution by solving the momentum, continuity, and Poisson pressure equations simultaneously.

A 1-D solution turns on the compressible momentum solver for 1-D subgrids. 1-D subgrids can be turned on simultaneously with the 3-D solver, thus two *Momentum lines may appear. The 1-D solver has two arguments, eps1d, and courant1d. ‘eps1d’ adds artificial viscosity to the 1-D compressible flow algorithm. The recommended value for eps1d is 0.5. The courant1d value can exceed 1, in fact there is no stability limit for courant1d so the user is free to chose any suitable value. However large time steps rapidly increase the amount of numerical dispersion in 1-D momentum solutions.

Uwm

Fraction upwind differencing in the X, Y, or Z direction. The recommended value is zero, which will result in full 2nd order accurate central difference.

xwm

Fraction crosswind differencing in the perpendicular X, Y, or Z directions. The recommended value is 0.1, which will result in 90% central difference and 10% upwind differencing. A value of zero is valid and will result in a full second order accurate term but may have stability problems with some problems.

Ncorr

The number of elliptic solver iterations in solving the pressure corrector equation. The recommended value is 12. Larger values may be required to problems with initial transients. It is recommended that values less than 50 be used in order to keep the code fast running.

Errmx

The maximum allowable equation of state error in pressure iterations. The parameter is problem specific and may require experimentation by the user. Typically, a value 0.001 for problems on the order of 10 meters in scale, and 0.0001 for problems less than a meter in scale will result in suitable solutions.

Sor

The successive over relaxation parameter for the momentum equation. The recommended value is 1.5.

Gx, Gy, Gz

Gravity in the X, Y, and Z directions.

*Energy Type uwe iter sor

*Energy

A keyword that turns on energy equation solvers.

Type

One equation (1eqn) or two equation (2eqn) solutions are options. A 1eqn solution has a single temperature in each cell. A 2eqn solution will have two temperatures (structure and fluid) in each cell.

Uwe

The fraction upwind differencing for the energy equation. Recommended values range between 0.2 and 1.0 – depending upon the degree of jaggedness the user is willing to tolerate in the solution. Smaller values approach second order accuracy but tends to have spurious numerical oscillations in parts of the solution where steep temperature gradients occur.

Iter

The number of elliptic solver iterations in solving the energy corrector equation. The recommended value is 10.

Sor

Successive over-relaxation parameter for the energy equation. The recommended value is 1.5.

*Species Name uws iter sor MW [3d-flag , 1d-flag]

*Species

A keyword that turns on species equation solvers.

Name

The name of the species equation. An ASCII string surrounded by quotes such as 'fuel.' The **specie name is case sensitive** and is used in identifying which specie equation to use in chemical reactions, and material injection from 1-D subgrids.

Uws

Fraction upwinding in the species equation. Zero implies full central difference and one results in full upwind differencing. The recommended value ranges from 0.2 to 1.0.

Iter

The number of elliptic solver iterations in solving the species corrector equation. The recommended value is 10.

Sor

The successive over relaxation parameter for the species equation. The recommended value is 1.5.

MW

The molecular weight of the species.

[3d , 1d, and conserve Flags]

'On' or 'Off' flags for turning on and off the species equations in the 3-D and 1-D subgrids. The user can choose to turn on or off the species equations so that materials, which may be created by chemical reactions will be tracked. Alternatively, the user can turn off specie equation solutions for materials that have no impact upon the solution such as inert materials. The flags are optional, if they do not appear in the file, then the code will assign the flags based upon the presence of the various species in the initial conditions.

The conserve flag forces conservation of chemical species by multiplying the current concentration at any location by a loss scale factor. The code uses a segregated solver to generate the solution to the system of equations. A segregated solver means that each equation, momentum, energy, species, is solved in series. They are not solved simultaneously during a single time step, thus the solution from one equation is not affected by the solution to the other equations. However, the solutions do affect each other at the next time step. This non-synchrony causes a slight error in species conservation. Although the error is small, some problems may require strict tracking of the total amount of species in the domain. When this option is turned on, a scale factor multiplies the final concentration of a species to force total conservation.

### 6.4.2 Chemical Reaction Equations

*Chemistry*

*Chemistry

A keyword which turns on chemical reaction equation input.

nreac

The number of chemical reactions to enter.

type

The chemical kinetics model. There are four options: Combustion, Sootmodel, Arrhenius, and Phase Change.

The next set of lines repeats for reactants and products.

Nreac

The number of chemical reactants.

CoefR

The mass coefficient of reaction. The coefficient of reaction is mass based because the species equations are mass rather than mole fraction based.

SpecR

The species name of the chemical reactant. The name must have the same spelling, and case sensitivity, as the name specified in the species equation.

Coef and Name are repeated for nreac reactant.

Nprod

The number of chemical products.

CoefP

The mass coefficient for chemical products. The sum of chemical product coefficients must be equal to the sum of reactant products if the masses are to balance. The code will not check mass balance due to chemical reaction.

SpecP

The name of the chemical product. Null is a valid name and it tells the code not to track the production or transport of chemical reaction products. The name is case sensitive.

Pre-Exp

The pre-exponential coefficient for the kinetic rate of chemical reaction.

E/R

The activation energy divided by the gas constant. The units must be in K.

Hc

The energy yield (heat of combustion in example) of the chemical reaction in J/Kg.

Tmin

The minimum temperature for the chemical reaction. The code will only calculate chemical reaction if the fluid temperature in the computational cell exceeds Tmin.

### 6.4.3 Soot Model and Hydrocarbon Gas Absorption

The soot model is used in calculating the total heat release from the fire, and in calculating the diffusive radiation parameters. The heat release from the fire is governed by the theoretical maximum (which is computed from the heat of combustion), less the heat lost due to radiation, and further decreased by chemical reaction inefficiencies. The chemical reaction inefficiencies are calculated during the run by the chemical reaction models. There is no default soot model thus the user must define the chemical species and reactions in the soot model. The chemical reactions and soot parameters in the example input file have been adjusted to approximate existing experimental data.

The soot model is activated by the following lines:

```
c       Fv_min   emiss   tinf    view   nvapors   fuel-oxygen-soot
*Soot    .5e-6     1     477     'on'      3     'fuel'  'o2'  'soot'
c name  curve scale   name   curve scale   name  curve   scale
 'fuel' 'CH4'   5    'intr' 'CH4'   2    'soot' 'SOOT'  5.55
```

*soot

This keyword turns on the soot model.

Fv_min

Fv_min is the critical volume fraction of soot that defines the surface of the fire.  Values of soot volume fraction greater than Fv_min are inside the flame zone, and values less that Fv_min are outside. A typical value for Fv_min is 0.5 PPM. Cells within the flame zone use the soot model to define the diffusive radiation conductivity. Cells outside the flame zone use view factor radiation to transport heat.

emiss

The gas emissivity of the fire surface cells.

tinf

The ultimate heat sink temperature to which the fire surface cells radiate.

view

This character string turns on, or off, the clear air surface radiation model.

nvvapors

This parameter defines the number of gaseous species to use when calculating the Rosseland mean absorption length for diffusive radiation. If nvapors > 0 then the second line above must be entered.

Fuel-oxygen-soot

The next three parameters are the specie names for the fuel species, oxygen species, and soot species. This will allow the code to identify which species will or will not be present in a fuel fire vapor dome. The critical soot volume fraction that defines the outer surface of a fire can also be satisfied in the vapor dome of a fire, which is lean in soot but rich in fuel vapor. By entering the names of the species the code will check what species are present, if fuel vapor is present in a greater concentration than oxygen, then the code will assume that it is in the vapor dome and will not include that computational cell in the fire surface definition.

For nvapors > 0, the user must enter nvapors sets of names and scale factors that correlate a built in radiation absorption model with a species name. nvapors must be less than or equal to the number of species. The first name is the species name, (same name used when defining species). The second name selects which correlation to use in evaluating the extinction coefficient. There are six possible choices:  Soot, H2O, CO2, CH4, PC_CH4, and PC_CH2.  The choices PC_CH4 and PC_CH2 are products of combustion, i.e. mole fraction interpolations of the $H_2O$ and $CO_2$ absorption curves. The interpolations are based upon the chemical balance formulas for burning methane (PC_CH4) and long hydrocarbon chains with an approximate hydrogen to carbon atom ratio of 2 (PC_CH2 – for example diesel fuel). If the user wants to model a species other than shown, the closest choices should be made. The scale factors allow the user to multiply the default extinction coefficient by a scale factor. In this way, if experimental data is obtained, a scale factor could be entered which will allow the built in curves to match the experimental data. For example laser extinction measurements at Sandia National Laboratories (Murphy and Shaddix, 2003) have produced data that indicate a scale factor of 5.5 for the relationship between the extinction coefficient for soot and its volume fraction.

*Diff_reac

This parameter is a proportionality coefficient that multiplies the effective reaction time when using the eddy dissipation combustion model. The turbulent time scale is added to the chemical kinetics reaction time to create a net reaction time scale, which is used as a source term in the species and energy equations.

*End

This keyword turns off the equation input.

## 6.5 Modeling Parameters, Material Properties, and Correlations for Heat and Momentum Transfer

This section allows the user to set up modeling parameters that may be used in defining the problem. It includes initial conditions, heat and momentum transfer correlations, material and fluid properties, turbulence model parameters, and mass and energy source term parameters. The user will select which properties, correlations, etc. to use when defining the problem in the next (geometry) section by the use of pointers to the definitions in this section.

```
###########################################################################
*Regions
#
*Fluid       Type        w   gam  pres0    name
            'igas'      29   1.4  0.82e5   'air'
*GasProp
            'Cond'  'Radiation'   1       .1
            'Visc'  'Constant'  2.e-5
            'Cond'  'PwrSeries' .035 0.001 -0.0001 0.0
            'Cp'    'spec_temp' 'air' 3  300   1000
                                         1580  1250
                                         2000  1350
                               'fuel' 2  300   2200
                                         2000  2200
                                'o2'  3  300   1000
                                         1580  1250
                                         2000  1350
                               'soot' 2  300   2000
                                         2000  2000
                               'intr' 2  300   2200
                                         2000  2200
                             'oil_pc' 2  300   1100
                                         2000  1500
*GasReg   #   press  temp  rux   rvy  ruz   m1   m2   m3   m4  m5
```

```
          1   .82e5  300    0.    0    0     0  0.233 0    0   0
*Struc #   Key   temp   Kg/m3    K    c    M2/m3 emis hext tinf alpha  ‘name’          mass frac
       1  'Con'   300.  803.     0   1e5      0    0   0    0     1   'isolator'       0 0 0 0 0
       2  'Con'   450.  803.    .9   1e5      0   .5   0   477    1   'fuel1'          0 0 0 0 0
       3  'Con'   300.  803.     0   1e5      0    0   0    0     1   'pool1'          0 0 0 0 0
       4  'Con'   300.  803.     0   1e5      0    0   0    0     1   'Air Supply'     0 0 0 0 0
       5  'Con'   300. 6000.    25   300e6    0   .95  0   477    1   'HGF Face'       0 0 0 0 0
*Power  Reg     Key       power(m),period(m),pdelay(m),tpwroff(m),fgw(m)
           1   'Exp'     1e5      .001     0.0         1e10       1
*Heat   Reg     Key         a *Ren **B * Pr**c + d   Hdia
           1  'h'    5
*Drag   Reg     Key   eps  Hdia
           1  'Moody'  .001  .5
           2  'moody'  .03   .5
*Turb   Reg    Key     Type       coef  (ax (x - x0) + by (y-y0) + cz ( z-z0) ) **d   +  e
           1  'Eddy'  'LES2'      .05    0      0     .1     2.8   0      0      1     .001
*Orfice  Reg   Key      %Areax   Coefx  %Areay Coefy  %Areaz Coefz
           1   'Normal'    0       1       0      0      0     1
           2   'Normal'    0       0       0      0      0     1
*end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*Regions

This keyword turns on the modeling parameters definitions part of the input.

### 6.5.1 Fluid Property Definitions

The fluid properties and initial conditions are defined here. Only a single fluid can be defined which can be either a liquid with Boussinesque density approximations or an incompressible gas. A compressible gas option is currently inactive. The primary difference between the two fluid types is the equation of state. For the liquid, the density is constant, with a slight linear temperature dependency. The incompressible gas satisfies the gas equation of state, where density is a function of temperature and mean molecular weight.

The definition of the type of fluid follows the format:

*Fluid Type   Parameters

*Fluid

This keyword tells the code that a fluid definition follows.

Type

This defines the fluid type. There are three options, 'Igas,' 'Gas,' and 'Liquid.'  Igas is an incompressible gas, Gas refers to a compressible gas, and Liquid is an incompressible liquid. Igas satisfies the gas equation of state with a mean pressure that may vary in time but not with position. Igas uses an incompressible flow solver. Gas uses the gas equation of state and a compressible flow solver, thus the pressure can vary with position and time (this option is currently not fully implemented). Liquid uses a constant density (incompressible) solution with Boussinesque approximation for the equation of state.

Parameters for different fluid types

The next three parameters depend upon the type of fluid defined:

| | |
|---|---|
| If a compressible gas | W, Gam |
| If an incompressible gas | W, Gam, Press0 |
| If an incompressible liquid | W, Rhol Beta, Temp0 |

W

Molecular weight of the background fluid. Note that when species equations are included additional molecular weights are included in the equations section of the input file. The background fluid is defined as the molecular weight of the fluid that has a mass fraction 1 minus the sum of all the other species at any location of the problem. Thus, a user need only solve one specie equation for problems with two species, because the background fluid is assumed to occupy the space not occupied by the species being tracked.

If the fluid is Igas, (incompressible gas) then the following two arguments must be input.

Gam

The ratio of specific heats of the fluid Cp/Cv.

Pres0

The mean pressure of the incompressible fluid.

If the fluid is 'Gas' (compressible gas) then one additional argument follows

Gam

The ratio of specific heats of the fluid.

If the fluid is Liquid (incompressible, constant density fluid with Boussinesque equation of state), then three arguments must follow.

Rhol

The density of the fluid.

Beta

The volumetric expansion coefficient of the fluid.

Temp0

The temperature at which the fluid has density Rhol.

Name

The last entry is the name of the fluid. This name must not be the same as a specie name. The name is used in establishing fluid identity when evaluating specie-dependent thermophysical properties.

*GasProp Type

*GasProp tell the code to read the thermophysical property data for the fluid. There are at least three, or optionally five properties and they may be entered in any order. They are: the specific heat at constant pressure 'Cp,' the viscosity 'Visc,' the conductivity 'Cond,' Rosseland radiation conductivity [‘radiation’], and mass diffusivity [‘diff’]. Two possible forms of conductivity can be entered: normal thermal conductivity and radiation diffusion conductivity. Both forms can be activated simultaneously. In the input file the property being defined is denoted by the appropriate keyword in quotes. The rule to be used in calculating the property is defined by any of the following keywords:

'Constant'

A constant value of the property data is entered.

'DATA'

A temperature vs. property data list, which is treated as piecewise continuous. The first line contains the number of data pairs plus the first pair of data. The first line is followed by the rest of the pairs of data. Each data pair consists of Temperature and Value.

'Spec_Temp'

A species and temperature vs. property data list, which is treated as piecewise continuous. The first line contains the species name, the number of data pairs, plus the first pair of data, followed by the rest of the pairs of data. Each data pair consists of temperature and property value. There must be nspecies + 1 entries. The first specie is the background gas (air in most cases). This is followed by the data for the rest of the species. The order of entry is important, they must be entered in the same order as the names of the specie equations.

'PwrSeries'

A power series curve fit of the form $a + bT + dT^2 + dT^3$ , where the user supplies the values a,b,c, and d. All four values must appear even if they are zero.

'PwrLaw'

A power law fit of the form ‘Value = a $T^b$.’ The user must supply the values a and b as two arguments and T is the temperature of the fluid in Kelvin.

'CurveFit'

A third degree (or less) polynomial curve fit to the data is used. The curve fit is exact so that the user must judge how well both interpolations and extrapolations will be over the temperature range of the problem. With this keyword defined, the code expects to read a number, which tells how many data pairs to read. A maximum of four data pairs is allowed. The data pairs consist of temperature and value.

'Chapman'

This keyword refers to the Chapman-Enskog theory for thermophysical gas properties. If this keyword is defined then two parameters must follow, they are:

sigma

The Lennard-Jones molecular cross section. These cross sections are printed in several text books for heat and mass transfer, such as Bird, Steward and Lightfoot (1960).

epsok

The molecular collision force parameter.

If the 'Chapman' keyword is chosen, the specific heat $C_p$ for the gas is calculated as:

$$C_p = \frac{R}{W\left(1 - 1/\gamma\right)}, \qquad (26)$$

where R is the universal gas constant (8314 in S.I. units), W is the molecular weight, and $\gamma$ is the ratio of specific heats. This formula is valid for perfect gases.

The initial conditions for the fluid are defined next. The user can define as many different fluid initial conditions as desired. They do not have to be used, and have no impact on the run time speed of the code.

*Gas Regions reg press temp rux rvy rwz m1 m2 m3....

*GasRegions
A keyword that turns on fluid initial conditions definitions.

reg
A pointer for the different gas regions.

Press
The initial gas pressure.

temp
The initial gas temperature.

rux
The initial x momentum $\rho u_x$.

rvy
The initial y momentum $\rho v_y$.

rwz
The initial z momentum $\rho w_z$.

m1, m2, m3....
Initial mass fraction of each chemical species in the fluid phase. There must be as many initial mass fractions as there are species equations or an error will result.

### 6.5.2 Structural Property Definitions

The user can define as many different types of structure as desired. The structure definitions do not have to be used and have no impact upon the run time speed of the code.

Structure property definitions follow the format shown below.

*Structure  #  Key Temp  Kg/m3  K   c    m2/m3 emis hext tinf  %solid mass-fractions

*Structure

A keyword that turns on structure definitions.

#

A pointer for the structure type definition.

Key

The structure type keyword.  There are two type keywords, 'Con' and 'Var.'  'Con' refers to constant properties, and 'Var' refers to variable thermophysical properties.

Temp

The initial temperature of the structure.

kg/m3

The density of the structure. The mass of structure in a cell is the product of its density and its volume fraction (%solid).

If the properties are constant (keyword ‘Con’) then enter the following two values:

K

The structure thermal conductivity. If a value of zero is entered then the structure will not transfer any heat to the fluid by conduction or to neighboring cells. However a value of zero does **not** turn off conduction, convection or radiation to 1-D subgrids.

C

The specific heat of the structure.

If the thermophysical properties are variable (keyword ‘VAR’) then there are three options then the material property can be a function of temperature (and possibly species). The next line contains two keywords and the first data pair.

Keyword 1 ‘CP’ for specific heat, or ‘TK’ for thermal conductivity.

Keyword 2 defines the type of property correlation. ‘DATA’ for a set of temperature-property data pairs, ‘POLY’ for a cubic polynomial, ‘CONS’ for a constant value, or ‘Spec_Temp’ for species and temperature dependent properties.

If keyword 2 is 'DATA' then enter number of data pairs (ndata) followed by the first data pair of temperature and property value. This is followed by ndata-1 lines of data pairs.

If the second keyword is 'poly,' then enter six numbers, the first four numbers are the coefficients of the cubic polynomial fit through the data, the last two are the max and min value respectively.

If the second keyword is 'CONST,' then enter the constant value of the property.

If the second keyword is 'Spec_Temp,' a species and temperature vs. property data list is entered, which is treated as piecewise continuous. The first line contains the species name, the number of data pairs, plus the first pair of data. This is followed by the remaining pairs of data. Each data pair consists of temperature and property value. There must be nspecies entries. This is followed by the data for the rest of the species. The order of entry is important, they must be entered in the same order as the names of the specie equations.

Next repeat the procedure for the second data set (CP or TK).

m2/m3

The surface area per unit volume for convective heat transfer to porous media. The surface area for heat transfer within a computational cell is determined by multiplying this parameter by the volume of the cell. A value of zero will turn off convective heat transfer. This surface to volume ratio is only used when porous media is being modeled. If the structure is 100% solid, the code will calculate the surface area internally. Note that heat transfer coefficients must be turned on in order to exchange energy.

Emis

A surface emissivity for the structure for radiative heat transfer to an ambient temperature, and to fire surfaces, see tinf below for ambient conditions.

Hinf

A heat transfer coefficient to an ambient temperature. See Tinf below.

Tinf

The ambient (environment) temperature for radiative and convective heat transfer defined above.

alpha

The volume fraction (0 to 1) of solid material in the structure. A value of 1 implies solid structure, which has zero flow velocities and gas densities. A value less than 1 is assumed to be porous structure with the balance of the cell volume being flow volume. This parameter affects the mass of structural material within a computational cell. The structure mass within a cell is determined by multiplying the material density ‘Kg/m3’ parameter times the cell volume times the ‘alpha’ parameter. This parameter also affects conductive heat transfer by its effect upon the cell-to-cell contact surface area. The surface area for conduction heat transfer within structures is assumed to be the product of the computational cell interfacial area times alpha. The contact area for gas to structure conduction is assumed to be the computational cell interface area times (1.0 - alpha). This parameter has no affect upon the specific heat of the structure or on convective heat transfer.

name

A name for the structure. The name must be surrounded by single quotes and no longer than 20 characters.

[M(I)]

The initial mass fractions of the species in the solid phase. These are optional input parameters. If they do not appear, all initial species mass fractions are set to zero. If the mass fractions are entered, there must be nspec entries, otherwise an error will occur.

### 6.5.3 Source Term Definitions

The source terms are defined here and allow the user to supply heat, and/or mass and chemical species to the problem as an internal source.

The format to follow for defining source terms is:

*Power Reg Key Parameters.... fgas [name]

*Power

A keyword that turns on parameters for source terms such as heat generation. Three basic time histories are assumed, a Gaussian shape, an exponentially rising/descending rate, and a table of values vs. time.

Reg

A pointer for a power region. If the pointer is **negative** then a spatially distributed power region is being modeled. If space dependence is modeled the code will open a file called profile.inp and will attempt to read a set of power distribution numbers. There may be more than one shape distribution, therefore each must be identified by a name. If a negative pointer is encountered the name must be input as an ASCII string at the end of the data line. The format for the file profile.inp is shown in the example below.

Example of a file profile.inp for providing space distributed shape data:

```
'Core'
'power'  14  17  1
.479  .489  .479  .480 …………………
.488  .490  .491  .492 …………………
```

The first entry 'Core' is a name that appears in the input file at the end of the data line. The second line indicates the type of distribution that is being modeled, followed by the number of X, Y, and Z computational cells. The code will check to make sure the correct number of data entries are in the file. The last line indicates the data, which is looped in the X, Y, and Z directions consecutively. The data in the file profile.inp is normalized to 1 such that:

$$\frac{\sum f_{(i,j)} V_{(i,j)}}{\sum V_{(i,j)}} = 1, \tag{27}$$

where the $f_{(i,j)}$ are user input values for the power distribution and $V_{(i,j)}$ are the volumes of the computational cells.

Key

The power time history keyword. There are currently five keywords for power. They are 'Gaussian,' 'Exp,' Square, and 'Table' and 'Mass.' Depending upon the keyword the code will expect to read a different numbers of arguments as explained below.

For the 'Gaussian' Keyword, the following parameters are expected:

Gaussian W/m3 dpulse period fgas

W/m3

The volumetric power generation rate in Watts per cubic meter. This parameter is input for all of the different types of keywords. If a distributed power region is being defined then the local average volumetric power is the product of the distribution shape function, and the W/m3 parameter.

Dpulse

The full-width half-max of a Gaussian power pulse.

Period

The time peak of the Gaussian pulse.

Fgas

The fraction of the power that is generated that is in the gas. The balance (1-fgas) is placed into the structure.

For 'Exp' keyword an exponential form is assumed and the following parameters must be input

Exp W/m3 period pwr-on pwr-off fgas

W/m3

The volumetric power generation rate in Watts/$m^3$.

Period

The exponential period of power up and power down. If a square pulse is desired then the periods can be set to a very small value, such as a microsecond.

Pwr-on

The time at which power is turned on full.

pwr-off

The time at which power is turned off.

fgas

The fraction of the power that is generated in the fluid. The balance (1-fgas) is placed into structure.

For the 'Square' keyword an square wave (on, off) form is created. The following parameters must be input:

Square W/m3 pwr-on pwr-off fgas

W/m3

The volumetric power generation rate in Watts/m$^3$.

Pwr-on

The time at which power is turned on full.

pwr-off

The time at which power is turned off.

fgas

The fraction of the power that is generated in the fluid. The balance (1-fgas) is deposited into any structure in the same computational cell.

For the 'Table' keyword a numerical table of values vs. time will be read from the file table.inp. For this keyword the following values are expected:

Table W/m3 ncol fgas

W/m3

The volumetric power generation rate in Watts/m$^3$. This parameter multiplies the power level read from the table.

Ncol

The data column number for the amplitude of the power. Note that the format for input in Table.inp is: time, column1, column2, column3,....

fgas

The fraction of the power that is generated in the gas. The balance (1-fgas) is placed into structure.

For the 'Mass' keyword a numerical table of mass source values vs. time will be read from the file table.inp. For this keyword the following values are expected:

Mass rate ncol temp

rate

The mass rate multiplier. This parameter multiplies the mass source rate read from the table. This allows the user to change the rate without rewriting the table.inp file.

ncol

The data column number for the magnitude of the mass source. Note that the format for input in Table.inp is time, column1, column2, column3,....

temp

The temperature of the mass source.

### 6.5.4 Heat Transfer Correlations

Heat transfer correlations are defined in this section of the input file. The user can define as many correlations as desired.

The format for assigning heat transfer correlations is:

*Heat # key parameters...

*Heat

A keyword that defines the heat transfer coefficients parameters and types.

# A pointer for a heat transfer correlation.

Key

There are several available key words as follows:

| | |
|---|---|
| ‘H’ | A fixed heat transfer coefficient |
| ‘Cyl_cross’ | A cylinder in cross flow |
| ‘Lam/Turb’ | A laminar & turbulent Nusselt/Reynolds correlation |
| ‘Wall’ | A flat plate correlation |
| ‘Stag’ | A stagnation point correlation |
| ‘Grash_[ ] | A Grashoff number correlation |
| 'Nusselt' | A Nusselt/Reynolds correlation |

Depending upon the keyword, a different number of parameters may be input as follows:

‘H’ Value

Input a constant value of heat transfer coefficient in W/m$^2$-K.

‘Cyl_cross’ Value Hdia

Where Value is the cylinder cross velocity and Hdia is the hydraulic diameter. If a cross velocity of zero is entered the code will use the locally computed velocity in calculating the heat transfer coefficient.

‘Wall’ Value Length

Where Value is the velocity across the flat plate and Length is the length of the plate. If a velocity of zero is entered the code will use the locally computed velocity in calculating the heat transfer coefficient.

‘Stag’ Value Length

Where Value is the normal velocity into the stagnation point and Length is the size of the stagnation point. If a velocity of zero is entered, then the code will use the locally computed velocity in the nearest computational cell when calculating the heat transfer coefficient.

‘Grash_[ ]’ Value Length

Where [ ] refers to either grash_h (horizontal) or grash_v (vertical) orientation. Value is the bulk temperature, Length is the length scale of the object. If a temperature of zero is entered the code will use the locally computed temperature in calculating the Grashoff number and heat transfer coefficient.

'Nusselt’ a b c d hdia

Input four parameters general heat transfer coefficient of the form Nu = a $Re^b$ $Pr^c$ + d, where

Nu = Nusselt Number,
Re = Reynolds Number, and
Pr = Prandtl Number.

Note that in evaluating the Reynolds number the local gas velocity is used. The superficial gas velocity is equal to the local velocity times (1.0-% solid). Thus, the user must adjust the coefficients ‘a’ and ‘b’ as required depending upon the definition of the heat transfer coefficient and its intended usage.

hdia

The Hydraulic diameter used in the Reynolds and Nusselt number definitions.

'Lam/Turb'

A combined laminar-turbulent formulation which consists of two lines of input. All of the data in the 'Nusselt' form shown above must be entered for the laminar flow region followed by a line of data for the turbulent flow region:

a,b,c,d,Hdia
a,b,c,d,Re-cr

a

The turbulent ‘a’ parameter for the standard heat transfer correlation.

b

The turbulent ‘b’ parameter for the standard heat transfer correlation.

c

The turbulent ‘c’ parameter for the standard heat transfer correlation.

d

The turbulent ‘d’ parameter for the standard heat transfer correlation.

Re-cr

The critical Reynolds number for transition to turbulence.

Hdia

The hydraulic diameter.

### 6.5.5 Momentum Drag Correlations

Momentum equation drag correlations are defined in this section of the code. These correlations are useful for porous media, or modeling pressure drops that the 3-D grid cannot resolve, but the user knows are there. They can also be used to predict the pressure drop in a pipeline, gap or other flow structure for which pressure drop correlations exist. The correlations are used in addition to any computational pressure drop that is predicted. Therefore, the model is optional. Consequently, it can be applied for calculations where a specific pressure drop relationship is required.

The format for pressure drop correlations is:

*Drag # Keyword Parameters

*Drag

A keyword that turns on drag coefficient models. Drag coefficients apply a flow resistance opposite the direction of flow. They are normally associated with porous media but the user can use these formulas to represent the type of flow resistance.

#

A pointer for the drag coefficient.

Key

The drag coefficient keyword. There are currently four different types of keywords for drag. They are:

| | |
|---|---|
| 'Reynolds' | Reynolds number dependent drag coefficient |
| 'Lam/Turb' | Reynolds number with transition to turbulence |
| 'Moody' | Moody diagram drag coefficient for internal flows |
| f(r) | Position dependent drag formula |
| 'DragCoef' | Constant drag coefficient |

For the keyword 'Reynolds' the drag formula is assumed to have the following form:

$$\Delta P = \frac{1}{2}\rho V^{2}\left[\left(a\,\mathrm{Re}^{b} + c\right)\frac{\Delta x}{h_{dia}}\right], \tag{28}$$

and the following parameters must be input.

a

The ‘a’ parameter in the above formula.

b

The ‘b’ parameter in the above formula.

c

The ‘c’ parameter in the above formula.

Hdia

The hydraulic diameter used in the drag formula.

Keep in mind that all of the velocities in the formula above are **local** velocities not superficial velocities, thus the constants must be defined appropriately. For example, the Ergun equation for pressure drop for flow through porous media is defined according to superficial gas velocities. However, since CAFE-3D is calculating friction based upon local flow velocities the values of a and c will be different.

For the 'Lam/Turb' Keyword the all of the data in the 'Reynolds' form shown above must be entered for the laminar flow region followed by a line of data for the turbulent flow region:

Re-cr

The critical Reynolds number for transition to turbulence.

a The turbulent ‘a’ parameter for the standard drag correlation.

b The turbulent ‘b’ parameter for the drag correlation.

c The turbulent ‘c’ parameter for the drag correlation.

For the keyword 'Moody' the drag formula is assumed to follow the Moody chart for pressure drop in pipe flow. The following parameters must be input.

ε

The relative roughness of the pipe wall, (m).

Hdia

The hydraulic diameter used in the drag formula (m).

For the keyword 'f(r)', the assumed drag formula is shown in the equation below:

$$\Delta P = \frac{1}{2}\rho V^2 \left[ \left( a \,\mathrm{Re}_x^b + c \right) \frac{\left( x - x_o \right)}{h_{dia}} \right], \tag{29}$$

where the Reynolds number is now position dependent. The following parameters must be input:

a,b,c,d

The a,b,c,d parameters in the above formula.

X0, Y0, Z0 The x, y, z position from which to measure distance. If X0 is zero then the position dependence is in the y, and z directions only. The same rule applies to the y and z directions. If the user wants to measure distance from the zero position, use a very small number (i.e., 1.e-10).

'Drag'

the drag formula for this option is shown in the equation below:

$$\Delta P = \frac{1}{2} \rho V^2 C_{df} , \tag{30}$$

where ΔP is the pressure lost in each computational cell in the flow direction and the following parameter is input.

$C_{df}$

The form loss coefficient for flow.

### 6.5.6 Turbulence Models

There are several turbulence models that the user can select for the problem being solved. All the available models are of the eddy diffusivity type. The user can select between an algebraic models, large eddy type models, or a one equation turbulence model – or any combination of these models. A two-equation model (such as k-epsilon) is not available. The algebraic models come in two forms, either as Reynolds number correlations, or simpler from where eddy diffusivity is an algebraic function of velocity and position. In the Reynolds number correlations, the eddy diffusivity is proportional to a Reynolds number based upon the local velocity and, either position in the domain, or a user assigned length scale. In the large eddy formulations, the eddy diffusivity is proportional to velocity gradients in the fluid and the length scale is a function of the computational cell size or an algebraic function of position. In the one equation formulation, a conservation equation for the turbulent kinetic energy is solved and the turbulence length scale is an algebraic function of position.

The format for defining the turbulence models is:

*Turb Pointer Key-1 Key-2 Parameters

*Turb

A keyword that turns on flow turbulence models.

Pointer

A number defining the turbulence model.

Key

A keyword that describes the type of turbulence model. Currently the only keyword available is eddy viscosity, which is defined by the 'EDDY' keyword.

Type

The ASCII string defining the turbulence model type. There are several options and each option has a different number of associated parameters.

'Grid' six parameters

An algebraic turbulence model, which models the turbulence found downstream from a mesh or grid. Six turbulence parameters are expected for this type of model.

'Constant' one parameter

A single constant value of eddy diffusivity is input, units of $m^2/s$.

'Simple' one scale coefficient plus 8 position parameters

This model multiplies the scale coefficient by the flow speed and the length scale parameter (see below).

'1-Eqn' 10 parameters

A one-equation (kinetic energy transport) model of turbulence. Ten parameters are input, the scale coefficient, the dissipation coefficient, and eight length scale coefficients (see below).

'LES' 1 parameter

A large eddy model where eddy viscosity is proportional to the scale coefficient time velocity derivatives and a turbulence length scale (assumed to be the computational cell size). A single proportionality parameter is input.

'VLES' nine parameters

A very large eddy model where the first coefficient is a scale factor, and the next eight are an algebraic turbulence length scale (see below).

'Ren' four parameters

A Reynolds number formulation, where Reynolds number is calculated from the local velocity, viscosity, and an input length scale. Four parameters (A, B, C, and Hydraulic length) are input. The formula for eddy diffusivity is:

Eddy Diff= $\nu$ (A*Ren**B+C),

where $\nu$ is the fluid kinematic viscosity.

Many of the above turbulence models require an algebraic length scale definition. An algebraic length scale formula with eight parameters was chosen:

$$L=( a(x-x0) + b(y-y0) + c(z-z0) )^{b} + c,$$

where L is the turbulence length scale, a, b, c, x0, y0, z0 are constants and x, y, z are position. The constants x0, y0, z0 normally align with the fuel pool surface. This allows the turbulent length scale top grow with distance from the pool surface. Good agreement with fire data is found when the exponent b is equal to 3/2.

### 6.5.7 Orifice Definitions

The user can define multiple orifices to control where the flow can go. An orifice is defined to exist on the positive computational cell boundary anywhere within the computational domain. Thus, it can affect the flow in the X, Y, or Z direction. Orifices cannot exist inside the computational cell, nor can they have a non-orthogonal tilt. It has infinitesimal thickness so that it is useful for modeling thin walls, windows, doors, holes, or other structures. The pressure drop across the orifice is defined by the fraction open and a loss coefficient. A complete blockage is allowed, i.e. no flow across the cell boundary. The user can assign a definition in more than one direction, such as a blockage in the X, Y, and Z directions simultaneously. Orifices can be either fixed for the duration of the calculation, or they can vary in time to simulate opening doors windows or valves.

Heat transfer across an orifice can be modeled by a 1-D subgrid so that the user can place many materials within the orifice.

The format for defining orifices is:

*Orifice # Key Parameters [ %Areax, Coefx, %Areay, Coefy, %Areaz, Coefz]

*Orifice

A keyword that turns on orifice definitions.

#

A pointer for the orifice.

Key

Keyword for the orifice. There are two allowed keywords, 'Normal' and 'Time.' Normal refers to a simple orifice and Time refers to a time-dependent orifice such as a valve.

For the keyword 'Normal,' Input the following data pairs in all three principal directions:

%Areax

The face area ratio of the orifice on the x interface of the cell. If there is a structure on either side of the orifice then the ratio refers to the area ratio that is based upon the reduced flow area due to the structure.

Coefx

An orifice coefficient in the X direction.

Note that values of %area and Coef applies to the formulae for a pressure drop across a computational cell interface, which is:

$$\Delta P = \frac{Coef}{2}(\rho u)_{x,i,j} V \left( \frac{1}{ratio^2} - 1 \right), \tag{31}$$

Where the value of momentum flux and velocity are the **local** values at the i, j cell interface. If a structure is present on either side of the interface, then the mass flux and velocity will be increased by 1/(void-fraction) compared to the smeared value.

The %Area and Coef are repeated in the Y and Z directions. If a total blockage is desired then the %Area term is set to zero and Coef is set to a nonzero value.

‘Time’ N-Ax N-Cx N-Ay N-Cy N-Az N-Cz

For the orifice keyword ‘Time’ input the following:

N-Ax

The column number in the file Table.inp, which contains time dependent data for the normalized area of the orifice. The actual area ratio of the orifice is ratio (next parameter below) times the normalized time dependent area. A time-dependent orifice allows things such as the opening and closing valves, doors, and windows to be simulated.

N-Cx

The column pointer to the time dependent loss coefficient in file Table.inp.

These parameters are repeated for the Y and Z directions.

## 6.6 The 3-D Problem Description

With the definitions above (the modeling database) the user can define a grid and paste the model parameters and other definitions into the grid to define the overall 3-D problem.

Below is an example of a 20 × 10 × 20 rectangular domain with a 7 meter diameter circular pool fire at the bottom of the domain. The problem description includes parameters for creating the mesh, placing a vertical composite cylinder above the fire, and applying boundary conditions to the domain.

```
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
*object3d  'Fire'
*mesh
      'nxnynz'   0 0  0
      'search'
# x grid
      '-xpwr'   0      8.7554      12
       'xfix' 8.7554   8.9855       1
       'xfix' 8.9855  11.0145       6
       'xfix' 11.0145 11.2446       1
      '+xpwr' 11.2446    20        12
# z grid
      '-zpwr'   0      8.7554      12
       'zfix' 8.7554   8.9855       1
       'zfix' 8.9855  11.0145       6
       'zfix' 11.0145 11.2446       1
      '+zpwr' 11.2446    20        12
# y grid
       'yfix'     0    .3365        2
       'yvar'  .3365   4.4005       8
       'yfix' 4.4005   4.88315      1
       'yfix' 4.88315  5.11175      1
       'yfix' 5.11175  5.4516       2
       '+ypwr' 5.4516   10          8
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
*geom  type  xmin xmax  ymin ymax  zmin zmax             Gas Wall  Pwr  HT Turb  Drag  Orf
    'box'      0   20     0   10    0    20               1    0    0    1   1    0     0
c    ecyl  axis  x0   y0    z0  ymin   ymax  rady   radxz
'open_ecyl' 'y' 10 5.11175 10 5.1175 5.4516 .3341 1.0145  1    2    0    1   1    0     0
c    cyl    axis x0   z0   ymin     ymax     rad
'open_cyl'  'y'  10   10  4.88315  5.11175   1.0145       1    2    0    1   1    0     0
     'cyl'  'y'  10   10  .3365    4.88315   1.2446       1    2    0    1   1    0     0
     'cyl'  'y'  10   10  .0       .3365     1.0145       1    2    0    1   1    0     0
c  short cylinder to view pool location
     'cyl'  'y'  10   10  .0       .1        3.5          1    3    0    1   1    0     0
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
*Boundaries     Side         Type  start stop    start stop Arg1  Arg2  Arg3
               'a'          'flux'    0      0      0     0     0      300
               'a'          'conc'    0      0      0     0     0     0.23 0 0 0
               'left'       'flux'    0     10     0    20      1    300
               'right'      'hydro'   0     10     0    20    300
               'top'        'hydro'   0     20     0    20    300
c Radial  pool  xc   yc  rad  tboil, Hfg,    cp_fuel,  Rate0 Time0 Max  Min Rate
'rbot'  'evap'  10  10  3.5   450   3.00E+05  2200     0.072   2   0.25 0.01
'rbot'  'conc'  10      10      3.5     1  0  0  0  0
c
*end
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```

The Geomview code (or a VRML viewer) will allow the user to view the object and grid structure. The model above will create the object pictured in Figure 3. The fuel pool location is shown as the solid disk below the cylinder. When the code is executed, this disk would be removed from the model if the user assigned a fuel vapor injection boundary condition. Alternatively, the user could assign subgrids to the disk and inject fuel vapor directly into the domain from the disk at the bottom.

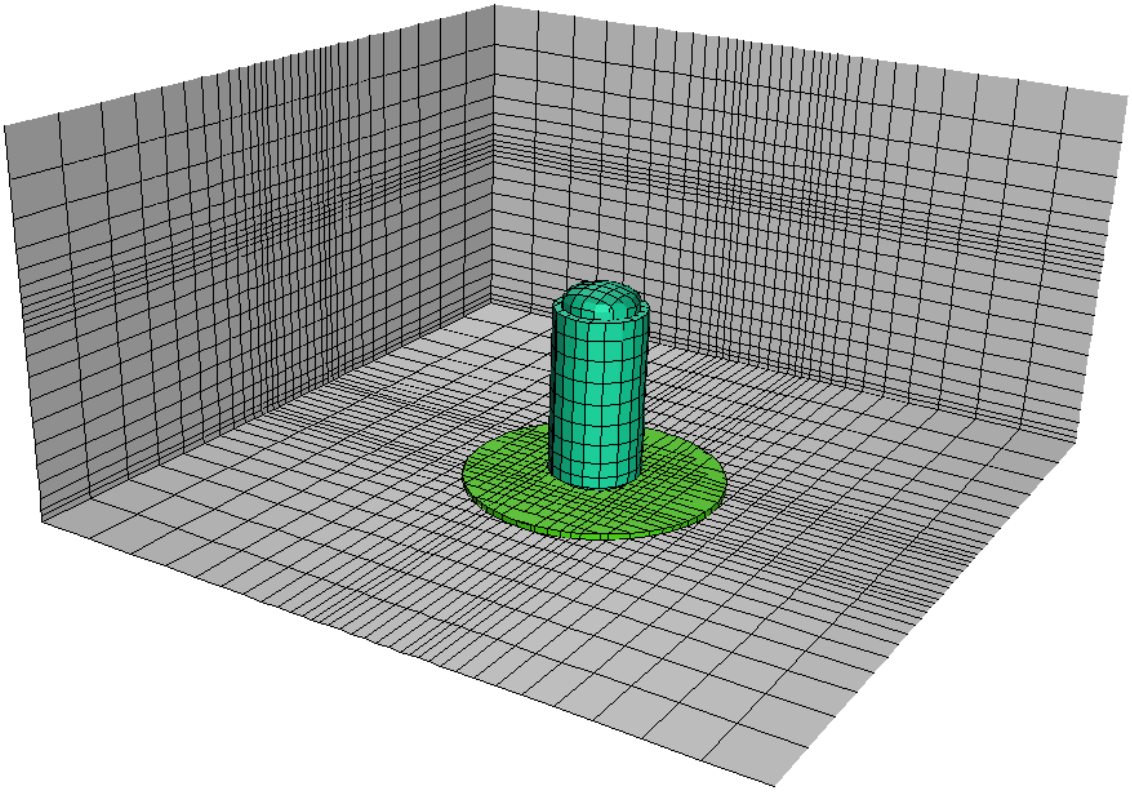

**Figure 3. A composite image created by the Geomview code showing the objects and the mesh structure simultaneously.**

The parameters for this section of the input file are defined below.

*object3d 'name'

This keyword tells the code that a 3-D object is to be defined. This control word is followed by an ASCII string that defines the name of the object. The 3-D problem definition section is terminated by a *end line. In the present version of CAFE-3D only a single 3-D object can be defined. Future versions will allow the user to define several different 3-D problems and couple the models together.

### 6.6.1 Grid Generation Commands

An example of mesh generation commands appears below.

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*mesh
      'nxnyz'    0 0  0
# x grid
      '-xpwr'   0      8.7554      12
       'xfix' 8.7554   8.9855       1
       'xfix' 8.9855  11.0145       6
       'xfix' 11.0145 11.2446       1
      '+xpwr' 11.2446    20        12
# z grid
```

```
      '-zpwr'   0      8.7554      12
       'zfix' 8.7554   8.9855       1
       'zfix' 8.9855  11.0145       6
       'zfix' 11.0145 11.2446       1
      '+zpwr' 11.2446    20        12
# y grid
       'yfix'     0    .3365        2
       'yvar'  .3365   4.4005       8
       'yfix' 4.4005   4.88315      1
       'yfix' 4.88315  5.11175      1
       'yfix' 5.11175  5.4516       2
       '+ypwr' 5.4516   10          8
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*mesh

Turns on the automatic mesh generation algorithm. The coordinate system in CAFE-3D consists of a box where the zero coordinate axis is located at the bottom-left-back corner of the three dimensional box. The x-axis runs from left to right, the y axis runs from bottom to top, and the z axis runs from back to front. The coordinate system is right handed.

The mesh generation algorithm will create a fixed mesh within the computational domain if no other commands are supplied. Typically, a user would want a refined mesh in some region of the problem and a coarse mesh in other regions. Creating a variable mesh requires the user to input additional mesh control parameters.

In the example shown above, the number of mesh cells and the size of the mesh are explicitly defined. The reason is so that the geometry is matched exactly with the cell edge boundaries. This is not a requirement, but it will increase accuracy when the objects are created.

'nxxynz'  30  40  50

These three parameters tells the code how many cells to assign in the X, Y, and Z directions. This command is the only required line in the mesh generator. A value of zero can be entered, however a zero instructs the code to add up the number of cells based upon the mesh generation commands that follow. A zero also requires the user to completely define the number of mesh cells in the chosen coordinate direction, which is the preferred option.

['xfix'], ['yfix'], ['zfix']

These optional parameters tell the mesh generation control algorithm to use fixed cell sizes over the specified region of interest. When using these parameters, three arguments must be input. The start position, end position and the number of computational cells to place in that region.

['xvar'], ['yvar'], ['zvar']

These optional parameters tell the mesh generation control algorithm to use variable cell sizes over the region of interest. When using these parameters, three arguments must be input. The start position, end position, and the number of computational cells to place in that region. The algorithm will try to fit a variable mesh geometry that varies smoothly from the start to end positions. When using this option, fixed cell sizes must exists in the surrounding regions.

[+xpwr], [-xpwr], [+ypwr], [-ypwr], [+zpwr], [-zpwr]

These optional parameters define a power law growth coordinate system. Three arguments must appear, start position, end position, and number of cells. A power law growth increases (or decreases) the cell size by a fixed multiplier. The multiplier is found by the segment length (end position minus start position) and the number of cells. To use this option, a boundary condition must be supplied, which is the cell size in an adjacent segment – thus one side of this option must include a 'fix' parameter. The adjacent side that has a fixed cell size is determined by the sign, a positive sign indicates that the cells will advance from the start position to the end position. A negative sign indicates that the cells will advance in the negative coordinate direction, i.e., from the end value to the start value.

When defining a mesh the user will use any combination of the 'xfix,' 'xvar,' '+xpwr,' and '-xpwr' commands. If the user does nothing other than *mesh and 'nxnynz' command, a uniform grid will be created in all three directions. By using a suitable combination of the var, fix, and pwr commands variable meshes can be created.

An alternate method of meshing exists where all cells are user defined. This method is the original mesh generation method in CAFE-3D and has been retained for backward compatibility and special problems where the user may want full control of the mesh generation process. It is recommended that this option be utilized by experienced users only. This section will be italicized to indicate that is an optional method.

*coord nxi nxf nyi nyf nzi nzf Type delx dely delz

#COORD

Is the number of lines to read for the coordinate set parameters.

NXI, NXF, NYI, NYF, NZI, NZF

Are the limits of the imbedded do loops over which computational cells to apply to the coordinate system definition. Note that this is cell number definition, not geometric position. Geometric position is calculated after the cells are defined, not before as in the previous coordinate generation algorithm.

TYPE

is the type of parameter being defined. There are several valid parameters, which are:

- DXDYDZ, in which uniform values of dx, dy, and dz are to be defined. This type of coordinate system should be defined first since the other parameters can overwrite or multiply the previously defined values.
- +XDX, -XDX, +YDY, -YDY, +ZDZ, -ZDZ define a power law growing coordinate system. The sign parameters determine whether the nodes grow in a

positive or negative X, Y, or Z direction.

- +DRDZDTH, -DRDZDTH is a cylindrical coordinate system. The user must input three variables, dr, dz, and dtheta. The + or - sign implies the + or - X direction for the radial part of the coordinate system.
- +ZDZX, -ZDZX, +ZDZY, -ZDZY parameters define an exponential growing dz (depth) coordinate in the positive (+), negative (-), X or Y direction.

VALUES

The arguments required for the coordinate system definitions. If dxdydz is defined then 3 values are required, they are the values of dx, dy and dz. If +xdx, +ydy, etc., are defined then a single value is input and that is the multiplier to use in calculating the coordinate system sizes.

### 6.6.2 Setting up the Problem

The next group of input lines describes the physical model or problem definition. The user must assign all modeling parameters based upon physical location, and can specify several types of geometric regions over which to apply the modeling parameters.

The procedure is to define a computational domain first. Next, solid object primitives can be overlaid on top of previously defined regions. All subsequent definitions replace existing ones in the same physical location. A gas region can replace a pre-existing solid region, thus holes can be created. In general, complex objects can be created by overlapping sets of geometric primitives. There are only a limited number of primitives available, so that very complex objects may be beyond the capabilities of the CAFE-3D solid modeler. Future versions of CAFE will allow the user to import a cad file (*.stl format) to overcome this limitation.

An example of setting up a physical model based upon geometric location is shown below.

```
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
*geom  type  xmin xmax  ymin ymax  zmin zmax             Gas Wall  Pwr  HT Turb  Drag  Orf
    'domain'   0   20     0   10    0    20               1    0    0    1   1    0     0
c    esph   axis x0   y0    z0  ymin   ymax  rady   radxz
'open_esph' 'y'  10 5.11175 10 5.1175 5.4516 .3341 1.0145 1    2    0    1   1    0     0
c     cyl   axis x0   z0   ymin     ymax     rad
'open_cyl'  'y'  10   10  4.88315  5.11175   1.0145       1    2    0    1   1    0     0
     'cyl'  'y'  10   10  .3365    4.88315   1.2446       1    2    0    1   1    0     0
     'cyl'  'y'  10   10  .0       .3365     1.0145       1    2    0    1   1    0     0
c  short cylinder to view pool location
     'cyl'  'y'  10   10  .0       .1        3.5          1    3    0    1   1    0     0
c
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
```

<u>*Geom 'type' xi xf yi yf zi zf Gas Wall Pwr HT Turb Drag Orf</u>

*Geom

This control card turns on the CAFE-3D solid modeler.

Type

This keyword defines the type of region. Currently there are eight possible types which are to be chosen by an ASCII keyword. The eight types are domain, box, cylinder, plane, plane eqn, sphere, equation, cone, enclosure, hole and wall (tube and ring were available in the previous version of CAFE-3D but the current version has not been updated to include these options). The Domain type can only appear once. It is used to set the size of the computational domain, objects that lie, or project beyond the domain are cut at the domain boundary. Box defines a 3-D rectangular region. Cylinder defines a smooth cylinder using the CPM approximation. Sphere defines a smooth spherical region. Plane defines a smooth planar surface. Equation defines a general 3-D elliptic equation. Cone defines a conical section. Enclosure creates a thin wall box with impermeable orifices as the walls. Wall creates a thin wall, and hole will cut a hole in the enclosure or wall definitions. All the objects defined above can only be aligned with one of the three principle axes, except the spherical, and plane regions. Rotation of an object into an arbitrary orientation is not yet available. Each geometric type requires different input as defined below.

There are also two cutting or clipping tools that can be applied to the geometric regions. These tools will allow the user to build more complex objects by defining a geometric shape then cutting and/or clipping it and then placing another object adjacent to it. The clip tool trims existing solid structures beyond the clipping region and fills the surfaces of the clip edge with CPM surfaces. The cut tool removes everything inside the cut region, (the opposite of the clip tool) and does not fill the ends with CPM surfaces. CPM surfaces will allow heat, mass, and momentum transfer to 1-D subgrids. To use a cut or clip tool, a geometric region is first defined (such as a cylinder), it can be followed by a cut or clip region or both. The cut or clip input line applies only to the geometric object immediately preceding, not to all geometric objects. The arguments to cut and clip are Xi, Xf, Yi, Yf, Zi, and Zf, which are the minimum and maximum in X, Y, and Z directions.

There are also two additional options for the object types: open_* and stair_*. where * is an object type (for example open_cyl). The open_ option does not fill in the CPM surfaces at the ends of the object. This option is used when building complex objects, for example multiple connected cylinders for example. The stair_ option follows the mesh cell boundaries and results in a stair step object – that is the same as a conventional finite difference object approximation. This option is useful in special circumstances where the intersection of multiple CPM objects could create undesired surfaces. Another use for the stair_ option is when code comparisons are being made to other finite difference codes that do not have the CPM approximation.

['Domain'] Xi Xf Yi Yf Zi Zf Model-Pointers

A box or rectangular region which requires three X, Y and Z pairs that define the minimum and maximum extent in each of the three principal X, Y, and Z directions. This defines the domain size, other objects that are defined but extend outside the domain will be clipped at the domain boundary. The domain is typically the first region that is defined.

'Box' Xi Xf Yi Yf Zi Zf Model-Pointers

A box or rectangular region which requires three X, Y and Z pairs that define the minimum and maximum extent in each of the three principal X, Y, and Z-Axis directions. All smaller box regions must reside within the principle domain.

'Enclosure' Xi Xf Yi Yf Zi Zf Model-Pointers

A rectangular enclosure with thin impermeable orifices as walls. This object requires 3 x, y and z pairs that define the minimum and maximum extent in each of the 3 principal X, Y, and Z-Axis directions.

'Wall' Xi Xf Yi Yf Zi Zf Model-Pointers

A single wall typically used with the enclosure objects. Wall is used to add additional walls to the model. The walls are modeled as thin and impermeable. This object requires three X, Y, and Z pairs that define the minimum and maximum extent in each of the three principal X, Y, and Z-Axis directions. Any two of the pairs can be identical, which defines the location of the wall.

'Hole' Xi Xf Yi Yf Zi Zf Model-Pointers

A single hole typically used with the enclosure or wall objects. Hole is used to add holes, windows doors to the model. A hole requires three X, Y, and Z pairs that define the minimum and maximum extent in each of the three principal X, Y, and Z-Axis directions. Any two of the pairs can be identical, which defines the location of the hole.

'Esph' 'Axis' X0 Y0 Z0 min max Rad1 Rad2 Model-Pointers

An elliptical sphere. The input consists of principle axis, the center of the cylinder followed by the minimum and maximum extent, followed by two radii of curvature. The elliptical sphere can be aligned with any of the three principal axes, X, Y, or Z. If aligned with the X-Axis, then enter the X, Y, Z coordinates of the elliptical sphere center, followed by the minimum X and maximum X extent (i.e. clip planes), followed by the X radius and Y-Z radius. If aligned with the Y-Axis, enter X, Y, Z coordinates of the elliptical sphere center, followed by the minimum Y and maximum Y extent, followed by the Y radius and X-Z radius. If aligned with the Z-Axis, enter X, Y, Z coordinates of the elliptical sphere center, followed by the minimum Z and maximum Z extent, followed by the Z radius and X-Y radius. An example of an elliptical sphere that is clipped by the min and max extent parameters is shown in Figure 4 below.

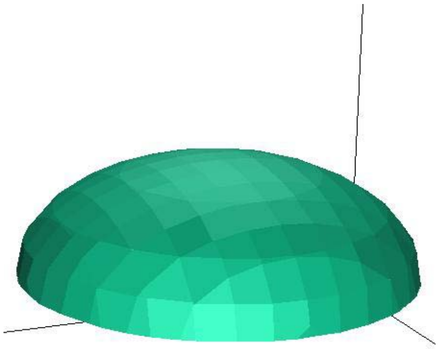

**Figure 4. An example of a clipped elliptical sphere with two radii of curvature created with the 'esph' command.**

'Cylinder' 'Axis' X0 Y0 Zi Zf Radius Model-Pointers

A Cylindrical region defines a smooth right circular cylinder. The input consists of principle axis, the center of the cylinder followed by the minimum and maximum extent, followed by the radius. The cylinder can be aligned with any of the three principal axes, X, Y, or Z. If aligned with the X-Axis, then the Y and Z coordinates of the cylinder center are entered, followed by the minimum X extent followed by the maximum X extent followed by the radius. If the cylinder is aligned with the Y-Axis, then enter X and Z positions for the center, Y min and Y max followed by the radius. If the cylinder is aligned with the Z-Axis, enter X and Y center coordinates followed by Zmin, Zmax, and radius.

'ECylinder' 'Axis' X0 Y0 Zi Zf Radius1 Radius2 Model-Pointers

To create an elliptical cylinder, input the following parameters. The principle axis, the center of the cylinder followed by the minimum and maximum extent, followed by the two radii of curvature. The elliptical cylinder can be aligned with any of the three principal axes, X, Y, or Z. If aligned with the X-Axis, then the Y and Z coordinates of the cylinder center are entered, followed by the minimum X extent followed by the maximum X extent followed by the Y radius and Z radius. If the cylinder is aligned with the Y-Axis, then enter X and Z positions for the center, Y min and Y max followed by the X radius and the Z radius. If the cylinder is aligned with the Z-Axis, enter X and Y center coordinates followed by Zmin, Zmax, and X radius and Y radius. An example of an elliptical cylinder is shown below in Figure 5.

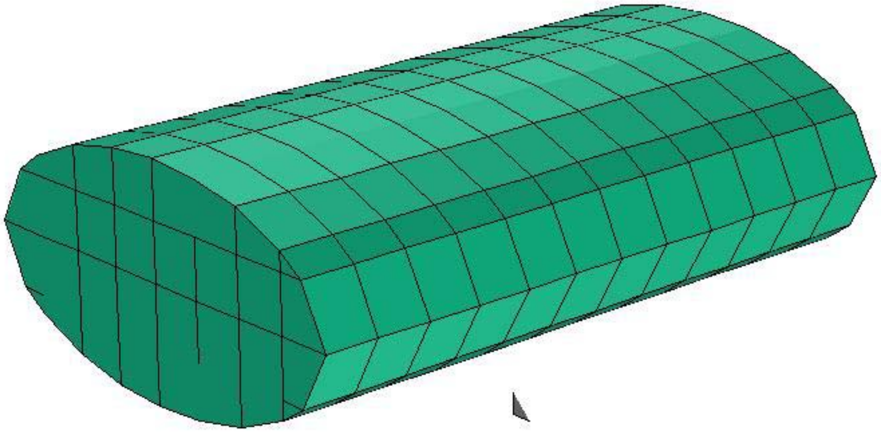

**Figure 5. An example of an elliptical cylinder made with the ecyl command.**

'Sph' X0 Y0 Z0 Radius Model-Pointers

'Sph' defines a smooth sphere. The arguments are center of the sphere at X0, Y0, Z0 and the radius.

'Plane_Eqn' A B C side Model-Pointers

'Plane_Eqn defines a smooth surface, which may be cut or clipped to make a wedge shaped object. The equation for a plane is Ax+By+Cz=1. The side parameter flips the geometric modeling parameters from to one side of the plane to the other side. If side is positive, the default side is used, if negative the other side is used. To view which side is appropriate for your model, run the code for a single time step and view the model with Geomview, or equivalent.

'Plane' xmin xmax ymin ymax zmin zmax side Model-Pointers
x1 y1 z1 x2 y2 z2 x3 y3 z3

This is identical to the plane object described above, except that two lines of input are entered. The first line contains the clipping box, the side of the plane which is solid, model parameters, and the second line includes three sets of points (x, y, z) that lie on the plane.

The side parameter is either a positive or negative 1 depending upon which side of the plane the solid material should be. Figure 6 below shows a box that has been clipped by a plane.

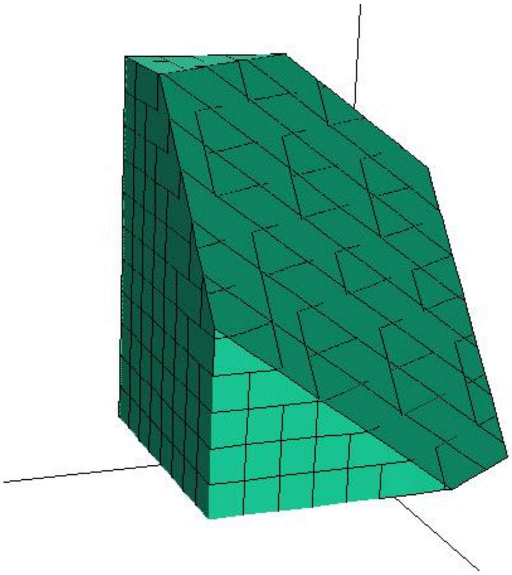

**Figure 6. A box that has been clipped by a plane. The object is easily created by the 'plane' object command.**

Only a single oblique plane can intersect the solid box, as shown in Figure 6. If the user desires multiple oblique planes, for example a wedge, then two boxes each with an oblique plane would be defined and placed adjacent to each other.

'Cone' 'Axis' x1 y1 z1 x2 y2 z2 min max Model –Pointers

When creating a cone, enter the axis of symmetry, followed by the focus of the cone (x1, y1, z1), and one additional point (x2, y2, z2) on the cone surface. Then enter the minimum and maximum extent along the cone axis. Finally enter the model pointers. An example of a cone object is shown in Figure 7.

### 6.6.3 Model Pointers

After the coordinate locations that define the shape of the region are entered then a set of pointers are defined which define which materials, and models to apply to those regions.

Gas

The pointer to the gas definition.

struc

The pointer to a structure definition.

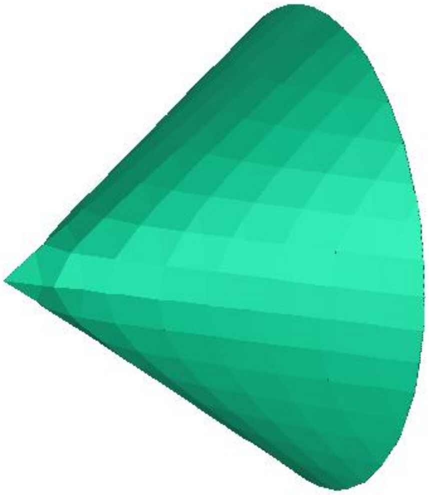

**Figure 7. A cone-shaped object created by the 'cone' command.**

Pwr

The pointer to a power definition.

Ht

The pointer to a heat transfer correlation.

Drag

The pointer to a drag correlation.

Orf

The pointer to an orifice definition.

The code uses the painters algorithm in defining the properties of the computational cells, i.e. a geometric region of nodes can be defined as having one set of properties and then overwritten later as having a different set of properties. In this way, complicated geometry's can be defined with a minimum of input lines provided that the user overlaps the regions in an appropriate manner. If smooth curved objects are defined and these objects are allowed to intersect, then it is possible to create unwanted holes at the intersection of the smooth curved surfaces and solid objects. Such holes must be filled by addition of solid material.

The values of the pointers can be zero, except that at least one gas or structure pointer must be nonzero. If the user is creating curved solid objects then both gas and solid pointers must be nonzero because the edges of the object will have both solid and gaseous regions. Additionally, if the structure is porous, then both gas and structure pointers must be nonzero. When the pointers are zero that model, correlation, or material property is turned off. For example if the structure pointer is set to zero, then that region will be filled with gas with the characteristics of the gas region pointer. If the heat transfer or drag pointers are set to zero then there will be no convective heat transfer or drag calculated. Pointers can be overwritten so that they can be defined on one line and then turned off by a subsequent line. This allows the user to create new regions with different properties within previously defined regions.

### 6.6.4 Assigning 3-D Domain Boundary Conditions

The boundary conditions (BC) are assigned after the model is built. The user can assign any combination of boundary conditions on any of the six faces of the computational domain. There is no restriction on the number or type of boundary conditions on any given face. The boundary conditions are assigned sequentially so that they can be overwritten by later definitions. Thus by overlaying sets of boundary conditions the user can create complex conditions to simulate many real world problems. Boundary condition can be independently assigned, or they may overwrite a previous definition. For example the user could assign a temperature definition on the bottom boundary, and later define a velocity boundary condition. These are independent because they apply to different governing equations. Certain boundary conditions are assigned to multiple equations simultaneously, such as the flux BC, which assigns both a normal momentum flux and a temperature. Such boundary conditions may overwrite previous definitions depending upon the sequence of BCs the user has assigned.

One common problem when running CFD calculations is an imbalance in inflow and outflow. If the total mass inlet flow does not balance the net mass outflow, problems may occur. If there is a net mass outflow from the system the total fluid density will decrease, temperatures will drop and the solution will eventually terminate with some type of error. The best way to insure continuity of mass is to use the exit and vexit type boundary conditions. Pressure BCs could be used as well but they do not converge as quickly as exit type BCs and may result is slower running problems.

In addition users must insure that the fluid zone is continuous and that there are no trapped bubbles of fluid within solids or fluid zones that are isolated from flowing boundaries. Trapped fluid bubbles or zones will result in very slow or nonconverging problems with small time steps.

<u>*Boundaries Side Type start stop start stop Arg1 Arg2 Arg3......</u>

*Boundaries

> This keyword turns on the boundary condition loop. As many boundary conditions are entered as the user wishes. The boundary condition loop is terminated by a *end statement

Side

> A character string that must be surrounded by quotes. It refers to the side of the domain over which to apply the boundary condition. The options are 'left,' 'right,' 'top,' 'bottom,' 'front,' 'back,' or 'all.' Optionally, the user can specify xmin, xmax, ymin, ymax, zmin, and zmax.

Type

> A character string surrounded by quotes which describes the type of boundary condition being applied. The various types are:

'flux' (two arguments)

A normal mass or momentum flux (i.e. kg/m$^2$-s) and temperature boundary condition. Two parameters are input, a normal mass flux and a temperature. The code will check if solid structure exists at the boundary and set the flow to zero for such conditions. If porous structure exists at the boundary then the mass flux applies in the void space of the computational node. If porous structure exists in the node at a boundary and the user wishes to specify a superficial mass flux at that boundary then the number that must be entered is the desired value of superficial mass flux divided by the void fraction of the computational cell. A negative value corresponds to the negative coordinate direction. This boundary condition is applied to both momentum and energy equations.

'mass' (two arguments)

An absolute mass flow rate in kg/s. Two arguments are input, the total mass flow rate, and the fluid temperature. This boundary conditions can be used in cases where the total mass flow rate is known, but the user may not know the flow area due to blockages etc. This boundary condition is applied to both momentum and energy equations.

'press' (two arguments)

A pressure boundary condition. The pressure is assigned in the center of the boundary node, not at the boundary interface. Two arguments are input, Pressure and temperature. This boundary condition is applied to both momentum and energy equations.

'time' (two arguments)

Read a table of values for the normal mass flux and temperature as a function of time from the file Table.inp. Two arguments are input, the column number for the normal mass flux, and the column number for the temperature. This boundary condition is applied to both momentum and energy equations.

'vel' (two arguments)

An explicit velocity boundary condition. It is explicit because the code uses the beginning time step value of gas density in generating the momentum boundary condition, and that value may slightly be different than the current end of time step density. Two arguments are input, the fluid normal velocity and the fluid temperature. This boundary condition is applied to both momentum and energy equations.

'table'

A normal mass flux boundary condition that has a specified shape as defined in the file profile.inp.

'Time_xyz' (3 arguments)

A 3-D mass flux boundary condition that has a time varying value as specified in file Table.inp. This includes all three momentum components hence it is particularly useful in applying time varying or wind boundary conditions. Three arguments are input which are the data column numbers for the x, y, and z components of momentum and column zero is always time. This boundary condition is only applied to the momentum equations.

'conc' (number-of-species arguments)
A species concentration boundary condition. The arguments are the mass fractions of all the species in the problem. An argument of zero implies a mass fraction of zero on the boundary, thus mass diffusion can occur and material can leave the system by diffusion. If an insulating boundary condition is desired input a negative number. This boundary condition is only applied to species equations.

'evap' (eight arguments)
A evaporative mass flux boundary condition that is based upon heat transfer induced evaporative boiling. This boundary condition is applied to both the momentum and energy equations. The mass fraction of the evaporating species is governed by the ‘conc’ boundary condition.

The arguments are:

| | | |
|---|---|---|
| 1. | Tboil | Liquid boiling temperature |
| 2. | Hfg | Latent heat of evaporation |
| 3. | Cp | Evaporating gas specific heat |
| 4. | Evap0 | Initial Evaporation rate |
| 5. | Time0 | Time over which to apply initial evaporation rate |
| 6. | Ev_max | Maximum evaporation rate |
| 7. | Ev_min | Minimum evaporation rate |
| 8. | Emiss | Surface emissivity/absorptivity for radiation |

'vgrad' (four arguments)
A normal mass flux boundary condition that has a specified gradient as defined by the addition of the four argument parameters. The gradient is defined by the sum of the normalized values of the start-stop parameters in the two orthogonal directions. This boundary condition applies only to the momentum equations.

'temp' (one argument)
A temperature boundary condition. This boundary condition applied to the energy equation only.

'hydro' (one argument)
A hydrostatic pressure boundary condition. This boundary condition can be used in place of the pressure BC where gravity head is included in the pressure term. The pressure at the coordinate axis is Press0. A single argument is required, the fluid temperature. This BC is applied to both momentum and energy equations.

'h' (two arguments)

A heat transfer coefficient BC. The heat transfer coefficient and the temperature of the boundary. This BC only applies to the energy equation.

'Exit' or 'Vexit'

An exit mass flux boundary condition that is computed internally based upon a mass balance over the entire domain. Exit is calculated once at problem startup. Vexit is calculated at the beginning of each time step. If there are mass sources or other time varying phenomena occurring in the problem, Vexit must be used to force total conservation of mass within the domain.

'vcross' (Two arguments)

A cross momentum boundary condition in the two orthogonal directions. Two arguments are required, the cross momentum components. This BC only applies to the momentum equations.

'wind' (three arguments)

A wind boundary condition that does not vary in time. Three arguments are required which are the X, Y, Z values of the momentum vector. This BC only applies to the momentum equations.

'rigid'

A rigid velocity boundary condition. This BC only applies to the momentum equations.

'free'

A free-slip momentum boundary condition. This BC only applies to the momentum equations.

start, stop, start stop,

The starting and ending physical position for the boundary condition. Two data pairs are required corresponding to the physical locations in the two coordinate directions.

Arg1, Arg2, ...

Arguments. Depending upon the boundary condition type these numbers can take on different meanings. See the definitions above for the values and numbers of arguments as function of BC type.

*end

This terminates the definition of the 3-D problem.

## 6.7 One-Dimensional Subgrid Problem Definition

The 1-D subgrids comprise a set of one-dimensional mass, momentum, energy, and species transport equations that can be coupled to, and transfer either vector or scalar field variables to and from the 3-D mesh. In essence, they allow the user to refine the mesh locally for heat, mass, momentum, and species transfer. The 1-D subgrids can be assigned by the user, created automatically, or both. The example below shows an automatic definition of a 20 node 1-D subgrid with adiabatic boundary conditions.

When running CAFE-3D standalone, 1-D subgrids are not required so that this entire section is optional. When running CAFE-3D linked to P/THERMAL, subgrids are required to pass information between the codes.

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*object1d  'cylinder'
*1Dauto  nxi  nxf               Gas Wall  Pwr  HT Turb Drag Orf   Len
          1    20                 1    2    0    0   0   0   0    .0254
*Boundaries  Side     Type   Arg1  Arg2  Arg3
            'left'   'flux'   0      0
            'right'  'flux'   0      0
*end
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

### 6.7.1 Defining 1-D Subgrids

[*Object1d]

This keyword turns on the definition of 1-D subgrids. 1-D subgrids is a method of local mesh refinement. In a 3-D problem with surfaces that may be thin, or where heat transfer needs to be resolved in greater detail, 1-D subgrids allow the user to resolve the fine scale conduction within the solid associated with the surface. The 1-D subgrids are not coupled to each other, thus there is no lateral heat conduction. Therefore, the heat transfer associated with 1-D subgrids will be accurate for conditions where the surface represents a thin object relative to the breadth of the surface, for example, a wall.

In a CAFE-3D calculation, the 1-D subgrids are used to pass information to the FE code. Heat transfer is not calculated for the 1-D subgrids but the surface temperature information is used in the 3-D simulation.

[*1duser]

This keyword turns on user defined 1-D subgrid models. When user defined 1-D subgrids are used, the total number of 1-D subgrids is assigned by the user. The user may wish to assign both user defined and automatic for some problems. User defined 1-D subgrids are useful for a variety of special purposes. For example, they can be used to simulate nozzles to inject flow (momentum) with a specified velocity, and chemical species to simulate a burner, or a fire suppressant.

[*1dauto]

This keyword turns on automatic 1-D subgrid models. In using the automatic 1-D subgrids, the number is assigned by CAFE-3D based upon 1-D coupling to surfaces in the 3-D problem. CAFE-3D will assign a single 1-D subgrid to each CPM surface within a 3-D cell.

nxi,nxf

These parameters control the 1-D distribution of materials and other modeling parameters set by the pointers. The modeling parameters are used within the node range nxi to nxf.

[nyi,nyf]

For user defined subgrids these parameters control the number of 1-D models. The user may wish to create the same 1-D model for many subgrids. When defining automatic 1-D subgrids, these parameters are not entered.

Modeling parameter pointers; Gas, Wall, Pwr, HT, Turb, Drag, Orf

These are Pointers to the same materials and physical model parameters as in the 3-D models described above.

Len

The physical length of the region set by nxi, nxf.

[Area]

The cross sectional area for nodes nxi to nxf. The area is required for user defined 1-D subgrids. For automatic 1-D subgrids the area is calculated by CAFE-3D based upon the surface area of the coupling computational cell.

### 6.7.2 Subgrid Boundary Conditions

*Boundaries

Turns on the definition of 1-D Subgrid boundary conditions.

Type

There are several types of boundary conditions that are allowed for 1-D subgrids. A list and definition of the different types appears below:

1 'Flux' A mass flux and temperature boundary condition. Two arguments mass flux (kg/m$^2$-s) and temperature (K).

2 'Temp' A fluid Temperature (K).

3 'H' A heat transfer coefficient (W/m$^2$-K).

4 'Hrad' Convective + radiative boundary condition to ambient conditions. 3 arguments, convection coefficient, surface emissivity, and ambient temperature.

5 'Conc' A species mass fraction, one value for each species in problem.

6 'Pres0' A pressure boundary condition (Pa).

7 'Pres3d' A pressure BC, where the pressure is local pressure in the 3-D grid. No arguments are required, the location is found from the coupling definition.

8 'Struc3d' A thermal boundary condition which transfers heat from the

subgrid to the associated structure. In order to be effective, the 3-D structure material must have a nonzero thermal conductivity. There are no arguments, the code assigns them according to the coupling definition.

9 'Time' A time varying mass flux (kg/m$^2$-s) and temperature (K) BC. Two arguments are the data column numbers from the file Table.inp.

10 'Hydro' A hydrostatic pressure boundary condition, the pressure is found from the hydrostatic head associated with the coupling location.

11 'Evap' fluid evaporation/boiling boundary condition. The fluid will evaporate based upon a heat balance at the boundary. The subgrid can receive heat from the 3-D main grid, and it can conduct heat into the interior of the subgrid, which may suppress or enhance the rate of evaporation. There are 7+nspec arguments. Boiling temperature, latent heat of evaporation, specific heat of evaporating species, initial evaporation rate, time span of initial evaporation rate, max local evaporation rate, minimum local evaporation rate, and the mass fractions of the nspec species that are evaporating. The surface emissivity is based upon the emissivity of the structural material of the surface cell.

Side

There are two sides to the subgrid: left and right.

Arg1, Arg2, ...

These are the arguments for the boundary conditions, which depend upon the type of BC. The number and value of arguments follow the same rules as for 3-D BCs.

## 6.8 Network Radiation

Network radiation is not yet fully implemented in CAFE-3D. What follows below is the network radiation model from CAFE-2D. When the 3-D version of network radiation is fully implemented, it will follow the same format and rules shown below.

```
###########################################################################
*rad   nxi  nxf   nyi  nyf  nzi  nzf  surface    radface    name   emiss   tinf  area
        20  20     5   21    1    1      1         'r'     'Refl-i'  .8     300
        22  22     5   21    1    1      2         'l'     'Core-i'  .8     300
        35  35     5   21    1    1      3         'r'     'Core-o'  .8     300
        38  38     5   21    1    1      4         'l'     'Refl-o'  .8     300
        47  47     5   21    1    1      5         'r'     'Vessel'  .8     300
        50  50     5   21    1    1      6         'l'    'Concrete' .8     300
*view
  'viewfactor'  'viewfac.inp'
###########################################################################
```

*rad nxi nxf nyi nyf nzi nzf surface radface name emiss tinf area

*rad is a keyword that turns on the network radiation models.

nxi, nxf,nyi,nyf,nzi,nzf

The node number limits over which to loop when defining radiation surfaces. All nodes within the loops are included as part of a single radiation surface.

surface

a number that identifies the surface. That number corresponds to the row and column number in the view factor matrix.

radface

the node surface which represents the radiating surface. The options are 'l,' 'r,' 'b,' 't,' 'k,' 'f,' and 'i,' which correspond to left, right, top, bottom, back, front, cell faces and internal. If ‘internal’ is defined then the user must input the radiating surface area and the code will partition the user-defined area by cellular volume weighting.

Name

An ASCII string of 12 or less characters that identifies the surface. The string is only used for user convenience in identifying the surface in the I3d.out file.

Emiss

The surface emissivity.

tinf

The background ambient temperature to which the surface will radiate if the sum of all other view factors is less that one. If the surface can see all other surfaces such that the sum of all view factors equals one then this parameter is not used.

Area

The radiating surface area per unit non-dimensional volume for surfaces that have an ‘i’ surface designation. Non-dimensional volume is defined as cell volume divided by the sum of all cell volumes in the radiating surface. If the radiating surface is not defined as ‘i,’ then the area parameter is not read by the code.

*view

A keyword that turns on the network view factor radiation input. The only available parameter is ‘viewfactor’ (see below). Additional parameters that tell the code to internally calculate view factors will be added in a future update of the code.

'viewfactor' 'viewfac.inp'

'viewfactor' tells the code that the filename containing view factor information follows. In the example the filename is ‘viewfac.inp.’

The format for viewfac.inp is shown in the example below:

```
*name  'Reactor'
6
'SPARSE'    1  2  3  4  5  6
1           0  1  0  0  0  0
2          -1 -1  0  0  0  0
```

```
3           0  0  0  1  0  0
4           0  0 -1 -1  0  0
5           0  0  0  0  0  1
6           0  0  0  0 -1 -1
```

The *name parameter identifies the radiation network name (see below under *object).

The next parameter assigns the number of radiation surfaces to read. This number must equal the number defined in the *rad section above.

The next string identifies the type of input to read. There are several options

1. 'Full,' the full matrix is input.
2. 'Lower,' the lower half of the matrix is input and the upper half is calculated from the reciprocity relationship.
3. 'Upper,' the upper half of the matrix is input and the upper half is calculated from the reciprocity relationship.
4. 'Sparse,' only certain elements are entered and others (those with negative values) are calculated from the reciprocity relationship.
5. 'Facet,' the FACET view factor code is used to calculate the view factors and the FACET output is pasted into the file.

Finally, the matrix is input. The first column of numbers correspond to surface number, the rest of the columns correspond to the view factors between surfaces.

## 6.9 Coupling between 3-D and 1-D Subgrids

Coupling of the 1-D subgrids to the main 3-D grid allows the user to model heat, mass, momentum, and species transfer on a highly refined 1-D mesh. This section describes how to assign the coupling between different parts of the 3-D model and the 1-D subgrids.

Heat, mass, momentum and species coupling to the 1-D subgrids occurs explicitly at the beginning of the time step. Stability issues are not normally encountered; however, it is possible to exceed stability limitations for cases that are tightly coupled. CAFE-3D tries to set a stable time step based upon explicitly calculated heat transfer parameters. However, if it fails, the user must control any instabilities by direct time step control - usually using a maximum time step size less than the unstable one will solve any stability issues.

Several examples are included in this section because of numerous options for coupling the meshes and each option requires slightly different format. Below each example, a definition of the various parameters is included. Parameters that have been previously defined will not be repeated.

The first example demonstrates the use of user-defined subgrids that are used to simulate nozzles injecting $CO_2$ at various locations and angles into the 3-D mesh. Two nozzles are specified in this example.

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*Coupling  'user'                     position        dir cosines
#
# CO2 Nozzles
#
#Surf typexfr1 name        region     x    y   z      xc    yc     zc   emiss
  1   'qms'    'Fire'      'point'  74.0 21.0 1.32   0.0   0.0   -1.000 1.0
#                Arg1       Arg2     Arg3       Arg4
               'Nozzle'   '1du'    'right'       1
  2   'qms'    'Fire'      'point'  70.3 21.0 1.32   0.0   0.0   -1.000 1.0
               'Nozzle'   '1du'    'right'       2
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

### 6.9.1 User Defined Coupling Models

*Coupling [user] , [auto]

This keyword turns on the coupling section of the code. There are two possible types: user-defined and automatic. If both types are to be activated, the user-defined coupling must be assigned first.

#Surf type name region x y z xc yc zc emiss

#Surf

A counter for the user defined coupling model. There can be an unlimited number of coupling models.

Type

The type of transfer between the 3-D and 1-D subgrids. There are three choices; 'q' = heat only, 'qm' = heat and mass/momentum, and 'qms' = heat, mass/momentum, and species transfer.

name

The name of the 3-D object for coupling. In future versions of CAFE-3D, there may be more than one 3-D model to couple.

region

The location of coupling. There are two options: 'point' and 'box.' Point assigns the coupling to a geometric point in the 3-D grid. This type of coupling is used to inject or withdraw material and/or heat to or from the 3-D grid. Box assigns the coupling to a computational cell face, group of faces, or a structural surface depending upon the next keyword. In this version, user-defined 'box' coupling is not fully implemented.

'point'

Coupling requires seven additional arguments. The arguments are, the 3-D x, y, and z point location, and xcos, ycos, zcos direction cosines of the momentum injection vector, and finally the surface emissivity, if there is a structure associated with the point. An emissivity must be entered even if there is no structure, the number will not be used in such a case.

The next line contains coupling information about the 1-D grid that is connected to the point injection. There are four arguments:

Arg1

The name of the 1-D model. There may be several 1-D models to choose from.

Arg1

The type of 1-D coupling. In this case '1du', 1-D user, is the only option. Future versions may allow other types of coupling.

Arg3

The 1-D boundary to couple to. There are two options: right and left.

Arg4

The number of the 1-D subgrid to connect to. The user can set up as many 1-D user subgrids as desired.

### 6.9.2 Automatic Subgrid Coupling

The next example shows three types of automatic subgrid coupling. Automatic subgrid coupling does the work of setting up the coupling network when there are many locations to couple to. For example modeling heat transfer to a wall or other solid object that has a large number of computational cells associated with is done automatically where the user specifies which groups of cells/subgrids to connect.

In the example below, there are two automatically coupled models. The first model couples a solid orifice structure (which simulates a thin wall, or ceiling) to a group of 1-D subgrids that will calculate the heat and mass transfer associated with a burning wall. The second model, couples a solid object, to another group of 1-D subgrids for calculating heat transfer within the solid object.

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*Coupling  'auto'
c model type     name      region  face    xi xf  yi yf  zi zf
  1    'qms'  'fire'       'box'   'orf'    1  30   1 21   1 30
              'wall'       '1da'            1  20
  2    'q'     'fire'      'box'   'g-s1'   1  30   1 21   1 30
              'cylinder'   '1da'   'left'   1
*end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

Automatic coupling is activated by the keyword 'auto' on the *coupling command line. The format for automatic coupling is described below:

| model | typexfr | name | region | face1 | xi | xf | yi | yf | zi | zf |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 'q' | 'fire' | 'box' | 'orf' | 1 | 30 | 1 | 21 | 1 | 30 |
|  | 'wall' | '1da' |  |  | 1 | 20 |  |  |  |  |

model

The first parameter is the number of automatic coupling zones to being defined. There may be several objects, or surfaces that have different characteristics. Each automatic coupling group can be assigned its own 1-D subgrid group, with a unique set of properties and dimensions.

Type

The type of transfer between the 3-D and 1-D subgrids. There are three choices; 'q' = heat only, 'qm' = heat and mass/momentum, and 'qms' = heat, mass/momentum, and species transfer.

Name

This is the name of the predefined 3-D. Currently only a single 3-D object can be defined, however in future versions of CAFE-3D multiple 3-D objects can be defined.

Region

This defines the geometric region over which to search for surfaces to couple to. 'Box,' is the only valid type at present and it defines a 3-D rectangular shaped. Other choices such as 'sphere' or 'cylinder' may be implemented in future versions.

Face

This word associates either a computational cell face or a structure surface for the coupling. If a cell face is chosen, the options are left, right, top, bot, front, and back. The code will pick the surface area associated with the cell face and assign it to the 1-D subgrid. If g-s[1] is chosen, the structure pointer associated with the number in brackets [ ] will be connected to the 1-D subgrids. By choosing a structure pointer, the user can construct an object with different structural materials, and each one would be associated with a different set of 1-D subgrids. If the object is constructed using the FAVOR method, then the diagonal surface will be chosen (if applicable). Optionally, the user can choose the name of the structure, and the code will search for any surface built out of that material with the chosen name.

xi, xf, yi, yf, zi, zf

These are the geometric limits over which to search the 3-D mesh for coupling surfaces.

The second line associates a 1-D subgrid to the 3-D cells defined by the first line.

Name

This is the name of the 1-D subgrid model to couple to. There may be several 1-D models with different names.

region

This defines the type of subgrid cells to associate the 3-D region with. There are two options, 1-D user defined, [1du] and 1-D automatic [1da].

If '1du' (user defined) is chosen, then it must be assigned to a single 1-D subgrid. This is a way to group a number of 3-D cells into a single 1-D subgrid. The 1du option will reset the cross sectional area of the pre-defined 1-D mesh to the appropriate size associated with the group of 3-D mesh cells. Currently the '1du' option is not fully implemented.

Face

The boundary of the 1-D subgrid, which is associated with the coupling, two choices left of right. **Exception:** If 'orf' is chosen in the previous line, then both left and right boundaries are used and are coupled automatically to the adjoining 3-D mesh cells. No parameter entry is needed. The 'orf' definition allows heat to transfer through a thin wall and communicate with the other side.

Cell

The computational cell number associated with the coupling. If the 'orf' definition is used, then two cell numbers are needed, one for each boundary.

## 6.10 Output Requests

The last section of the input file allows the user to control the types and amount of output that the code generates. There are several options for output control, each one can be plotted by different types of graphics packages. In the examples that follow an example and description of each type of output are provided.

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c----------------Output Parameters--------------------------------------------------------
*Output
c---------------Tecplot-------------------------------------------------------------------
*TEC3D delay   dt  xi  xf  yi  yf   zi  zf
         0     .1   0  35   0  27     0  32
c---------------GMV-----------------------------------------------------------------------
*GMV   delay   dt  xi  xf   yi  yf   zi  zf
         .0   .05   0  60   0   25    0  60
c-------------ACSII Formatted USER Defined subgrids---------------------------------------
*Ascii_1du     delay   dtplot ndat    i      j
               0.0      .1    2       3      8
                                      1      6
c-------------ASCII Formatted at x,y,z positions------------------------------------------
*Ascii      pdelay dtplot  ngroup
             0.0    .05      1
            'PoolCenter'    5   x     y     z
                                0   3.04    0
                                0   3.29    0
                                0   4.29    0
                                0   4.79    0
                                0   6.79    0
c-------------ASCII Formatted AUTO 1-D subgrids at specified x,y,z coupling locations-----
*Ascii_1d   pdelay dtplot  ngroup
             0.0    .05     3
               'WallHFG'    4   x     y      z   1d
                              -2.75  2.85    0    5
                              -2.75  3.85    0    5
                              -2.75  4.85    0    5
                              -2.75  5.85    0    5
               'PoolHFG'    4
                              1.015  2.85    0    1
                              1.015  3.85    0    1
                              1.015  4.85    0    1
                              1.015  5.85    0    1
               'Calorim'    4
                               .00   3.94    0    1
                              -.16   4.10    0    1
                               .00   4.26    0    1
                               .16   4.10    0    1
c-------------Binary XPlot 1-D user subgrids distribution---------------------------------
*1DU        delay    dtdist  #    start  nstop  1dU
             0.0      .2     3     1     10     1
                                   1     10     12
                                   1     10     15
c--------------Binary XPlot Auto 1-D subgrid distribution at specified x-y-z location-----
*1D       delay    dtdist   #     x      y     z   start stop
           0.0       1      4   39.45  .61    30    1    20
                                39.45  1.83   30    1    20
                                39.45  3.00   30    1    20
                                39.45  4.00   30    1    20
c------Angular Surface distribution  Temperature and Heat Flux ---------------------------
c*Surf1d  = Binary XPlot formatted
c*Surf1d_ASCIII  = ASCII formatted
*Surf1d     delay   dt   number   axis    xmin   xmax    ymin  ymax
              0     .5     1       'z'     0       6      2      4
c--------------Binary XPlot line distribution in 3-D grid parallel to x, y, or z axis-----
*Binary-1D # delay  dtdist   #   start  nstop   I2  I3  dir
               0       .1    2      2      6    26  15  'y'  ‘flame zone’
                                    2      6    26  16  'y'  ‘edge zone’
c--------------Binary XPlot 2-d contour slices or projections of 3-D grid-----------------
*Binary-2D   #   delay  dt
             4   0.0   .5
#Binary 2D  filename      axis   start  stop  start   stop  plane
             'pl2dz.bin'    'z'     0     30    0      30    15   ‘specz.bin’
             'pl2dy.bin'    'y'     0     30    0      30     0   ‘specy.bin’
             'pl2dy2.bin'   'y'     0     30    0      30     3   ‘specy2.bin
             'pl2dx.bin'    'x'     0     30    0      30    15   ‘specx.bin’
*end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*Output

A keyword that turns on output control

### 6.10.1 Debug Output

[*Debug  [dir]]

An optional keyword that turns on enhanced output. Without the *debug keyword, smaller and fewer output files will be created. The additional files that are created with *debug include:

Alpha.out; which includes an ASCII list in table format of geometric parameters associated with structure such as porosity, X, Y, and Z directional areas for flow and heat conduction. The optional parameter defines the marching direction for the output.

Facearea.off is a color coded .off file for viewing with Geomview. It gives the user a graphical 3-D display of the surface area for heat and mass transfer of the 1-D auto assigned subgrids.

Areax1.off, Areayi.off, Areazi.off are color coded .off files for viewing with Geomview. They give a 3-D graphical display of the flow areas in X, Y, and Z directions respectively.

### 6.10.2 GMV File Output

The General Mesh Viewer (GMV) software is a 3-D scientific visualization tool that can be used to render isosurfaces and contours of field variables generated by CAFE. Multiple time-sequenced GMV files can be used to generate movies. GMV files are ASCII files that can be transported between different computer platforms. Figure 8 is an example of an image generated in GMV illustrating a fire by plotting an isosurface of soot colored by temperature. For more information about GMV and learn how to obtain this freely available software, visit http://www-xdiv.lanl.gov/XCM/gmv/GMVHome.html.

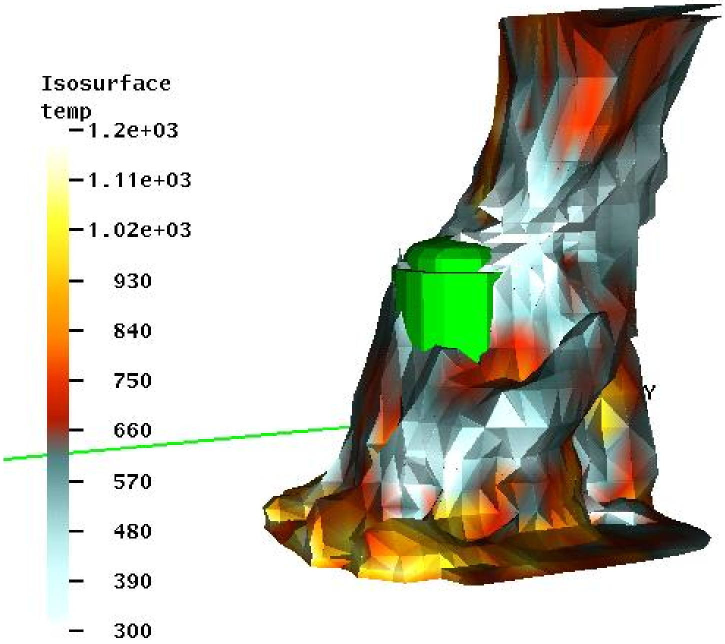

**Figure 8. GMV plot of soot isosurface colored by temperature. This figure shows a vertical cylinder partially engulfed in a pool fire with mild cross wind conditions.**

An example showing how to activate GMV output is shown below.

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*GMV   delay   dt  xi  xf  yi  yf   zi  zf  [ngmv]
         0     .1   1  35   1  27    1  32      0
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*GMV

Keyword that turns on GMV formatted output.

Delay

The delay time before which GMV output is produced.

dt

The time step between GMV output files.

xf, yi, yf, zi, zf

Defined the limits of the 3-D region where GMV output will be generated.

[ngmv]

This is an optional start number of the first gmv file. The gmv files are numerically sequenced so the user can create a movie. The last GMV file number generated is stored in the restart file, thus the user can continue the calculation. If a value exists for ngmv, that number will be used in the first gmv file that is produced. If no number is entered, then the first file will be named Temp001.gmv, or it will include whatever number is stored in the restart file, provided the code is run from a restart.

### 6.10.3 Tecplot Output

Tecplot is a commercial plotting and data visualization software that can be used to render isosurfaces and contours of field variables. Multiple time sequenced Tecplot files can be used to make movies. Tecplot files are ASCII so they can be transported between different machines. An example showing how to activate Tecplot output is shown below.

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*TEC3D delay   dt  xi  xf  yi  yf   zi  zf
         0     .1   0  35   0  27    0  32
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*TEC3D

Keyword that turns on Tecplot formatted output.

Delay

The delay time before which Tecplot output is produced.

dt

The time step between Tecplot output files.

xi, xf, yi, yf, zi, zf

Define the physical limits of the 3-D region where Tecplot output will be generated.

### 6.10.4 ASCII Text File Output

ASCII text file output will create a number of files with scalar, and vector field output in a large set of files with a *.out extension. An example activating ASCII text file output is shown below.

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*Ascii   pdelay dtplot  ngroup
           0.0    .5      2
            ‘first group’ 2     x      y      z
                              39.45   0     30
                              40.1    1     30
             ‘2nd group’  2
                              49.45   0     30
                              50.1    1     30
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

This type of output is arranged into groups to facilitate grouping together sets of data, for example thermocouple towers for comparison with experimental results. The grouped data will be included in the limits file outputs for Xplot.

*Ascii   delay   dtplot    ngroup
‘groupname’   ndat
 x y z
 x y z

*Ascii

A keyword that turns on ASCII formatted output for computational cell output.

delay

The delay time before which ASCII output data is printed to various output files.

dtplot

This is the amount of time between writes to the output files.

ngroup

The number of groups of data.

groupname

The name assigned by the user to the data group.

Ndat

Number of location (X,Y,Z ) sets for this group.

x,y,z...

The X, Y, and Z locations for producing ASCII output.  The X, Y, Z locations must have ndat entries. They cannot be on the same line.

Repeat for each group.

### 6.10.5 ASCII 1-D User Assigned Subgrid Output

This type of output will produce ASCII text files of selected 1-D, user-assigned subgrid computational cells. For example, the user may wish to examine the flow and temperature through a crack that is being simulated by a 1-D user subgrid. The format for 1-D user ASCII output is;

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*Ascii_1du     delay   dtplot ndat    i      j
               0.0      .1    3       3      8
                                      1      6
                                      1      10
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*Ascii_1du

A keyword to turn on ASCII output of field variables in 1-D user subgrids. The code will output: temperature, mass flow rate, and integrated species flow rate at selected 1-D user assigned computational cells.

delay

The delay time before which ASCII output data is printed to various output files.

dtplot

This is the amount of time between writes to the output files.

ndat

The number of locations for which the user wishes to have output ASCII output data.

i,j...

The cell number and subgrid number producing ASCII output.  The i,j data pairs must have ndat entries.

### 6.10.6 ASCII Auto Assigned 1-D Subgrid Output

The ASCII auto assigned 1-D subgrid output allows the user to generate text file output of field variables from automatically assigned subgrids. For example, if a cylinder is placed in a fire, and the detailed heat transfer within the cylinder is calculated by 1-D subgrids, the surface temperature, or inner wall temperature, can be output using this option.

An example of this output follows:

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*Ascii_1d   pdelay dtplot  ngroup
             0.0    .05     3
               'WallHFG'    4   x     y      z   1d
                              -2.75  2.85    0    5
                              -2.75  3.85    0    5
                              -2.75  4.85    0    5
                              -2.75  5.85    0    5
               'PoolHFG'    4
                              1.015  2.85    0    1
                              1.015  3.85    0    1
                              1.015  4.85    0    1
                              1.015  5.85    0    1
               'Calorim'    4
                               .00   3.94    0    1
                              -.16   4.10    0    1
```

```
                               .00   4.26    0    1
                               .16   4.10    0    1
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*Ascii   delay   dtplot    ndat          x y z, i  .....

*Ascii_1d

A keyword that turns on ASCII formatted output for a 1-D subgrid that has been automatically assigned.

delay

The delay time before which ASCII output data is printed to various output files.

dtplot

The amount of time between writes to the output files.

Ngroup

The number of output groups to follow.

Next for each group enter the following:

Name

The group name.

ndat

The number of locations for which the user wishes to have output ASCII output data.

x, y, z, 1-D

The 3-D X, Y, Z location(s) of the connected subgrid followed by the ith computational cell of the 1-D subgrid.  The x, y, z, 1d, data sets must have ndat entries.

### 6.10.7 Animated One-Dimensional Line Distributions within 3-D Model

When using Xplot, animated line (x-y) plots can be produced at any location in the 3-D computational mesh. The line distributions must co-align with one of the three principal axis directions. An example of this type of output request is shown below:

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*Binary-1D # delay  dtdist   #  start  nstop    I2   I3  dir
                0      .1    3   4     21.4    4.7  6.8  'x'  ‘name1’
                                 0     15.6    8.8  8.4  'y'  ‘name2’
                                 0     11.0    6.5  4.5  'z'  ‘name3’
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*Binary-1D # delay  dtdist  start  stop  I2 I3   Dir

*Binary-1d

A keyword to turn on output for 1-D field distributions anywhere in the 3-D mesh. The distributions must line in one of the principle (X, Y, or Z) axis directions. Diagonal distributions are not allowed.

delay

The delay time before output is written.

dtdist

The time between outputs.

start

The start position location.

stop

The stop position location.

I2, I3

The perpendicular axis positions of the distribution. Which perpendicular axis depends upon the next parameter, the axis direction. If X is chosen, enter the Y and Z locations. If Y is chosen, enter the X and Z location. If Z is chosen, enter the X and Y locations.

Dir

A single ASCII character 'X,' 'Y,' or 'Z,' which is the axis direction in which to write the distribution.

Name

The name of the distribution.

### 6.10.8 Animated Line Distributions for User Defined 1-D Subgrids

This option turns on Xplot binary formatted files for viewing animated line distributions of field variables in 1-D user defined subgrids. The example below shows the format for such an output request.

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*1DU        delay    dtdist   #    start  nstop n1d
             0.0      .2     8     1      10    1
                                   1      10    12
                                   1      10    15
                                   1      10    18
                                   1      10    20
                                   1      10    22
                                   1      10    23
                                   1      10    23
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

[*1DU], [*1DU_ASCII]

*1DU is a keyword that turns on output from 1-D user-defined subgrids.
*1DU turns on binary output for Xplot

*1DU_ASCII turns on the same output except in ASCII format.

Delay

The delay time between outputs.

dtdist

The time step between outputs.

#

The number of 1-D distributions.

start, stop

The beginning and ending node numbers for the distribution.

n1d

The nth 1-D user subgrid distribution to plot.

### 6.10.9 Animated Line Distributions for Auto Assigned 1-D Subgrids

This option turns on Xplot binary formatted files for viewing animated line distributions of field variables in 1-D automatically defined subgrids. For example, the user can view an animation of the temperature distribution within a cylinder placed in a fire, or view the parameters associated with thermal decomposition of wood as it burns. The example below shows the format for such an output request

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*1D         delay    dtdist   #     x    y   z  start  stop
             0.0      .2      2    4.3  10.2 1    1     10
                                   2.5   6   12   1     30
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

[*1D], [*1D_ASCII]

*1DU is a keyword that turns on field variable distribution from 1-D automatic defined subgrids. *1D turns on binary output for Xplot. *1D_ASCII turns on the same output except in ASCII format.

Delay

The delay time between outputs.

dtdist

The time step between outputs.

#

The number of 1-D distributions.

x,y,z

The 3-D x, y, z cell locations associated with the 1-D automatic subgrid for which to plot a distribution.

Start, Stop

The beginning and ending node numbers of the distribution to plot.

### 6.10.10 Animated 2-D Contour Plots within the 3-D Model, Slices and Projections

The user can request animated contour plot output of field variables along any slice parallel to one of the three principle axis, or, a contour plot of a maximum, or minimum projection within a volume. An example of such an output request follows.

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
*Binary-2D   Num   delay  dt
             4     0.0    .5
#Binary-2D  filename      axis   start  stop start  stop   location
              'pl2dz.bin'    'z'     1    35    1     30    16      'sp2dz.bin'
              'pl2dzp.bin'   'zp'    1    35    1     30     9  24  'sp2dzp.bin'
              'pl2dx.bin'    'x'     1    30    1     32    17
              'pl2dxp.bin'   'xp'    1    30    1     32     1  35
*end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
```

*Binary 2D

A keyword to turn on 2-D binary contour plot output for Xplot.

Num

The number of contour plot lines to read.

Delay

The delay time before which output is produced.

dt

The time step between outputs.

filename

The name specified by the user for the 2-D output data file.

Axis

The perpendicular axis to the contour surface over which the output will be produced. Which axis is chosen determines what the next entries are. There are nine possible choices: X, Y, Z are contour slices across the X, Y, and Z axes. xp, yp, or zp are contour projections over a positional range. The contour projection contains the maximum value at any location along a line of sight. xpm, ypm, and zpm are contour minimum value projections. xa, ya, and za are field variable time averages over the selected dt, this option is useful if the problem fluctuates and the user is interested in an average value.

xi, xf, yi, yf

The start and stop limits of the computational cells over which contour output will be produced. If x-axis was chosen, enter the y-start, y-stop, and z-start, z-stop locations. If y-axis was chosen, enter the x-start, x-stop and z-start, z-stop locations. If z is chosen enter the x-start, x-stop and y-start, y-stop locations.

[Location], [Start, Stop]

The perpendicular axis physical position where the 2-D contour output will be generated. If the axis is 'xp,' 'yp,' 'zp,' 'xm,' 'ym,' or 'zm,' then two values must be entered which are the start and stop positions of the axis over which the maximum (or minimum) value will be searched for.

[filename-2]

An optional file name, which, if present, will contain species data at the same contour slice.

## 6.11 Other Input files

There may be other input files that CAFE-3D reads for supplemental information. Currently only time dependent data is available in CAFE. Two or 3-D distribution data, and view factor data are not fully implemented.

Time dependent data is read from a file called Table.inp. The format for Table.inp is simple. On the first line enter the number of columns (N) to read. The second line begins the data. The first column (i.e. column 0) contains the time in seconds. This is followed by N-1 data columns (columns 1...ndat in I3d.inp). The data columns can represent anything the user wants. The I3d.inp file directs the code to which data column to read for the required data. The code will linearly interpolate between time lines in the file. It is highly recommended that the user enter a last line that exceeds the stop time of the calculation, otherwise the code will use the last value computed.

This page intentionally left blank.

# 7. Run time control and output

While the code is running, the user can extract output or exert other control functions whenever the code writes information to the screen. During each screen print the code will search the working directory for certain filenames. If such a file is present, it will trigger either a control function or an output file. The user can create a file with either a text editor, copy command, or if the user is on a UNIX platform, an empty file can be created with the “touch” command.

If the file is a request for a Geomview file (*.off) and the file is empty, the code will write the file using its own internal scaling. If the file contains a max and a min value, the code will scale the output (color coded) by the values contained in the file.

The filenames that are searched for are:

I3d_stop

This file will terminate the problem normally at the next screen output. Normal termination may include writing a restart file.

I3d_kill

This file will kill the problem at the next screen output without writing a restart file. This is a useful option if the user wants to preserve the data in the restart file.

Restart

This file will trigger the writing of a restart file at the next screen output. The restart file is called restart.inp.

GMV

This file will trigger the writing of the next gmv file. The user must have selected the GMV option for output because the code needs certain parameters in order to write the file.

TEC

This file will trigger the writing of the next Tecplot file. The user must have selected the TEC option for output because it needs certain parameters in order to write the file.

StrucTemp

This file will trigger the writing of a StrucTemp.off file for viewing with Geomview. The file is color coded and internally scaled by the maximum and minimum values of the auto assigned 1-D subgrids. This file displays the surface temperature of the coupled object. This file can be rescaled by user-assigned values if the file ‘StrucTemp’ contains a maximum and minimum value.

Aerosol

This file will trigger the writing of a Aerosol.off file for viewing with Geomview. The file is color coded and internally scaled by the maximum and minimum values of the auto assigned 1-D subgrids. This file displays the deposited density of aerosol on the coupled object. This file can be rescaled by user assigned values if the file 'Aerosol' contains a maximum and minimum value.

Speed

This file will trigger the writing of a Speed.off file for viewing with Geomview. The file is color coded and internally scaled by the maximum and minimum values of the surface of auto assigned 1-D subgrids. This file displays the gas speed on the surface of the coupled object. This file can be rescaled by user assigned values if the file 'Speed' contains a maximum and minimum value.

Tgas

This file will trigger the writing of a Tgas.off file for viewing with Geomview. The file is color coded and internally scaled by the maximum and minimum values on the surface of the auto assigned 1-D subgrids. This file displays the gas temperature on the surface of the coupled object. This file can be rescaled by user assigned values if the file contains a maximum and minimum value.

## 7.1 Troubleshooting Run Time Problems

When running the CAFE code the user may experience run time difficulties, for example the code may not run or may run very slowly. Fixing such problems depends upon the user's ability to diagnose the problem.

**Startup Problems**

*Problem: The* ***code will not run at all.***

Solution

If the code will not run at all, the problem is caused by a bad or missing input file. Normally the code will print a message indicating the problem. If the code doesn't print a message, it probably couldn't read the input file. In this case, examine the output that was printed, if it failed reading the input, the problem lies somewhere near or at the line that was printed last. The output normally is printed on the screen, if that option is not available, open file I3d.out in the directory where the code was executed.

If the code printed an error message it will indicate the type of error. Some Common Errors at Startup are:

Bad coupling area

This is caused by assigning a 1-D subgrid to some material or geometry in the main 3-D grid and the code failed to find that geometry or material. A common problem is miss-assignment of material number, or a poorly defined search box. Make sure that the search box includes the object that is being searched for.

Can’t find 1-D subgrid

This occurs when the names of 1-D objects were not consistent. Check spelling and remember that names are case sensitive.

Can’t read restart file

This type of error is caused whenever the user is reading a restart, but changes the input file in such a way that the number of computational cells change, the number of 1-D subgrids changes, a new solid appears where a fluid used to be, or vice versa. Typically these problems occur when the user changes the geometry, or mesh, and tries to read a restart. The solution is to rerun the problem from scratch.

Undefined temperature

The code initializes the entire memory with an undefined temperature. After defining the mesh and geometry, if an undefined temperature is encountered this means that the mesh size and geometry are not consistent. The user may have defined a mesh that exceeds the computational domain size, or vice versa. Check the boundaries of your mesh and geometry (primarily domain size).

Another possibility is that the user has defined a geometry and has not assigned a fluid to that geometry. If the geometry has curved surfaces (sphere, cylinder, etc.) computational cells on the boundaries must contain both solid and fluid. If no fluid is assigned, undefined temperature errors will occur. Check the solid and fluid parameters for all your geometric primitives.

Boundary Condition out of range

This error occurs when you have assigned a boundary condition that extends beyond the computational domain boundaries. Check the limits of your BCs and compare to the domain size.

When viewing a model - Geometry is defined but the object is not there or very distorted.

This can happen if you have created geometry and have not assigned a solid material to it. Check your geometry material parameters. Another possibility is that you do not have enough grid cells to resolve the object. This can make the object disappear or appear very distorted. In this case increase the mesh resolution (in all three directions) in the vicinity of the object.

**Run Time Problems**

Once the user has resolved all of the possible input problems and the code is running, it is still possible to have run-time problems. If the code runs and stops unexpectedly without an error message, check file I3d.out for errors.

Occasionally the code may have a run time error, such as divide by zero, which is captured by the operating system. Check material properties, especially variable properties, to determine if properties have become negative or zero. Normally this type of run time error is difficult to diagnose and should be reported to the code author, along with any debugger trace back information and the input file that caused the problem.

The most common run time problems fall into two main categories: negative temperature, and slow running.

*Problem: Negative temperature*

If this occurs, the computation has become unstable and terminated due to negative temperatures.

<u>Possible causes:</u>

The code has read an incompatible restart file and is attempting to execute it.

The input file may have bad thermophysical properties. If negative thermal conductivity, or zero (or negative) specific heat, is used in a computation, negative temperatures can result. Check thermophysical properties of all your materials, negative properties are not allowed. Zeros are allowed, except for molecular weight, pressure, density, and specific heat. If using variable properties, determine if they can become negative during the computation.

The possible cause is that the model has geometry or run-time parameters that ultimately cause a numerical instability. This type of problem can be diagnosed by examining the run time behavior. If the code runs with successively shorter time steps and larger errors, it is unstable and some parameters must be changed to enhance stability.

<u>Possible solutions:</u>

- Increase number of momentum iterations. If the code will not stabilize with 40 iterations this generally will not help.

- Change the mesh or geometry slightly. A very small cell flow area or volume can sometimes cause instability.

- Increase the amount crosswind differencing (xwm parameter) in the momentum equation solver.

- Check the values in the input file, make sure there are suitable values for all your input. Bad or nonphysical values of some parameters can cause problems.

*Problem: Slow Running*

If the problem runs slowly, i.e. a fraction of a millisecond time step, there could be several possible causes.

<u>Possible causes:</u>

- Pressure Oscillations. This type of behavior is often encountered in large domains that will support an oscillating pressure wave. A typical example is a long tunnel. It can also be caused when an enclosure separates two large volumes with some type of opening between them (i.e. a building with an open door). Diagnosing this type of problem consists of generating output such as animated 1-D line, or history plots at various locations in the domain. If pressure oscillations are encountered, they can generally be cured by reducing the max time step size, or by adding drag. This is especially true during problem startup when large pressure waves may be forced through the system due to the sudden onset of flow. After the problem is running smoothly, the max time step size can be increased.

- EOS (Equation of state) error control. This is generally the most common problem. It can be diagnosed by checking the run time control (istep) in the I3d.out file. If istep = 7, that is EOS error control. If this is the problem, check to see if the cell location causing the problem is fixed or if it varies. Cell location is printed out next to the time step error data. If the location is fixed, create a surface plot of flow speed, and EOS error. Check if the locations are co-located and if they are the same as the time step control location. Possible solutions are:

  - Modify the geometry and/or mesh slightly.

  - Make sure there are no isolated sections of the domain, or voids in solids. The fluid must be a continuous medium. Isolated voids within the domain or inside solid are not allowed. For example, a thin wall enclosure without an opening (door, window) is not allowed, since the interior would be isolated from the exterior.

  - Make sure the mesh has smooth transitions in all 3 (X, Y, Z) directions. The ratio between adjacent cells should not exceed 1.3.

  - Increase the amount of cross wind differencing (xwm parameter).

  - Increase the number of momentum iterations (ncor parameter).

  - Increase the amount of turbulent dissipation.

  - Increase the amount of momentum drag.

- Courant Number error control (istep=2 in the I3d.out file). This is caused by excessively high speed flow. Possible causes are:

  - The boundary conditions may be poorly defined, i.e., large pressure gradients or very large flow rates.

  - Courant number limitations may be an indicator of pressure oscillations (see above).

  - An incompatible mesh/geometry, a flow speed surface plot would diagnose this type of problem.

  - Another possibility is a net outflow condition. If boundary conditions are defined that have a net outflow, i.e., not enough inflow to conserve mass, the pressure in the domain will decrease, finally resulting in very high velocities due to the low gas densities. Monitor system pressure vs. time to detect this type of problem. The solution is to define a set of BCs that conserve mass.

# 8. Output File Description

The output from CAFE consists of six different types of files, ASCII, 1-D binary, and 2-D binary files, *.off files, *.tec files, and *.gmv files.

## 8.1 ASCII Output Data Files

The ASCII files are of three types a run time history of the calculation, simple time history data files suitable for any spread sheet plotting package and specially formatted graphics files for use with Xplot, Geomview, GMV, and Tecplot.

### 8.1.1 I3d.out

I3d.out, a problem setup and run time history file. I3d.out contains a reflection of the input data file, processed information about the model, error and warning messages, and a run-time history of the calculation. The error and warning messages are of great value when trying to figure out why the code will not run. Often times there could be a simple typo in the input, or not enough data on a line, or pointers or physical position out of bounds. I3d.out should be examined first to see what happened during the calculation if it fails to run to completion.

### 8.1.2 History Field-variable output data files

There are 15 types of simple data files with a file extension *.out. Within each file the first data column is always time, followed by multiple lines of data. The names of the data files are descriptive of the data contained in the file. The file names and contents are:

| | |
|---|---|
| Alpha.out | A number of flow and structure related parameters |
| Conc.out | The species concentration at user selected locations |
| Mass.out | A number of system summation parameters |
| Press.out | The fluid pressure |
| Qtot.out | Volume integrated heat generation and heat of chemical reaction |
| Qdot.out | Total heat generation/unit volume |
| Q_Xfr.out | Total heat energy transferred to all subgrids |
| Rho.out | Fluid density |
| Species.out | Total chemical species in computational domain |
| SpeciesBC.out | Total species inflow rate into domain |
| SpeciesSRC.out | Total species sources |
| SpeciesXFR.out | Total species transferred to subgrids |
| Temp_1d.out | Temperature history of selected subgrids |
| Temp_1d_ave.out | Average temperature of structure types in subgrids |
| Temp.out | Fluid temperature at selected locations |

### 8.1.3 Geomview output Files *.off

The ASCII graphical output files with a file extension *.off are intended to be used by the Geomview graphics code. Geomview is a three dimensional virtual reality viewer that is very useful for examining the CAFE model in 3-D. The file I3d.off will show the solid surfaces and orifices in the model. The various materials are color coded so that different materials and orifices can be identified. Grid.off shows the grid structure. A useful option is to open both I3d.off and grid.off simultaneously, this will display both the grid and structure (objects) at the same time. When using this mode be sure to turn on the 'show cell edges' option.

The file StrucTemp.off is a plot of the surface temperature of all solid objects in the model. The temperature is color coded and the scale varies between the minimum and maximum in the model. The values of the minimum and maximum temperatures are printed out in the I3d.out file.

The file Tgas.off is a plot of the gas surface temperature of all coupled solid objects in the model. The temperature is color coded and the scale varies between the minimum and maximum in the model. The values of the minimum and maximum temperatures are printed out in the I3d.out file.

The file Aerosol.off is a plot of the deposited aerosol density of all coupled solid objects in the model. The density is color coded and the scale varies between the minimum and maximum in the model. The values of the minimum and maximum temperatures are printed out in the I3d.out file.

The file Speed.off is a plot of the fluid speed on the surface of all coupled solid objects in the model. The speed is color coded and the scale varies between the minimum and maximum in the model. The values of the minimum and maximum temperatures are printed out in the I3d.out file.

### 8.1.4 GMV Output Files Temp*.gmv

CAFE-3D will output files called Temp*.gmv, where * is a numerical value. These files contain ASCII formatted data about the temperature and chemical species over the user selected portion of the mesh. The GMV graphics program can be used to view this data in 3-D. Normally a sequence of GMV files is produced during a run, and these files can be converted into a movie. A typical three dimensional image produced by GMV is shown in Figure 8.

## 8.2 Xplot Binary Data files for Animation

There are two types of binary data files that can be produced by CAFE: 1-D and 2-D. Both types are binary and can only be read by the Xplot viewer. They are intended to display a time animated sequence. The binary 1-D files will show one-dimensional distributions intersecting the 3-D grid at any location, parallel to any principle axis. If subgrid models are activated 1-D distributions of temperature, species concentration, velocity and other field variables can be viewed as well. The binary 2-D data files will display time animated 2-D contour slices along any pre-selected cut plane.

# 9. Changes to I3d.inp when running CAFE-3D

When CAFE-3D is running, there are a couple of changes to the I3d.inp file that the user needs to make. First of all, the user needs to run CAFE-3D standalone for a short period of physical time, say for example 10-20 seconds in order to engulf the object with flames. The restart parameter must be set to ‘w’ so that at the end of this preliminary run CAFE will write a restart file. After this preliminary ‘restart run’, the user needs to modify the I3d.inp file to read the restart file, by changing the rcmd parameter to 'r.'

Next, change the run (stop) time of the problem. P/THERMAL will have its own final stop time of the calculation. While P/THERMAL is running, CAFE-3D will be called frequently. The physical run time for CAFE-3D needs to be sufficient to allow the previous thermal history of the fire to be purged. Generally this takes approximately 200 milliseconds for objects that are on the order of 1 meter in size. A typical flow velocity in a fire is approximately from 5-20 m/s so that the user must estimate the physical run time of the problem from the size of the fire and object. If the user is modeling an enclosure fire, the situation is quite different. The user must decide if time scale acceleration (such as obtained by CAFE) is appropriate for the problem at hand. Some enclosure fires may require CAFE to run in real time in order to model the filling of the enclosure with smoke. Others can be run in a quasi-steady mode, such as those enclosures with large ventilation.

The output frequency of CAFE-3D for all types of output should be adjusted to the requirements of the user. In some cases, the user may not have access to the graphical tools that are supported by CAFE-3D, thus the delay time for the appropriate output types may be set to a very large number. This would prevent CAFE-3D from generating useless output and wasting disk space on files that could not be viewed.

This page intentionally left blank.

# 10. Summary

The CAFE computer code (Suo-Anttila et al., 1999) has been developed to model all relevant fire physics for massive objects engulfed in large fires. It provides realistic fire thermal boundary conditions for use in design of radioactive material packages and in risk-based transportation studies. The CAFE code is intended to be coupled with finite-element models of specific package designs. This coupled system can be used to determine the internal thermal response of engulfed objects to a range of fire environments.

This document provides a brief history of the CAFE-3D code and its predecessors. The document is a user manual which describes how to use CAFE-3D as a coupled code system, or CAFE in standalone mode. A description of all the possible input variables and types of output is included. Examples of the correct format structure in the input file are included. A description of the output files and the appropriate viewing software is described.

This page intentionally left blank.

# 11. Example Problem

The input file structure example shown in previous sections contained too many options for an example CAFE problem. In this section the input file for a working problem will be presented. The example problem contains many aspects that would be encountered in a real fire. The user can copy this file into a text editor and modify it to suit his or her needs. The example chosen is an enclosure pool fire with a cylindrical calorimeter positioned above the fuel pool. Variable fuel evaporation and the use of subgrids will be activated. This example is chosen because it is complex enough to cover a wide variety of modeling features in CAFE and can be easily modified to simulate a real world problem. For example, if an outdoor fire is desired, simply delete the enclosure and its openings, redefine the domain size and fire size, and remesh.

## 11.1 Example Problem Features

The example problem presented in this section has the following features:

1. A 3 meter pool fire, depressed 0.2 m below ground level.
2. A 1 m diameter, 3 m long, 1 cm thick, steel cylinder located 1 m above ground level, located at pool center.
3. An thin wall (2 mm steel) enclosure 10 m x 10 m x 5 m high.
4. A open door 4 m wide 3 m high.
5. A rear vent 2 m wide 1 m high at the rear ceiling.
6. A computational domain 20 m long (X), 10 m high (Y), 20 m wide (Z).
7. Active fire model with soot and radiation.
8. Variable evaporation.
9. 1-D subgrids for heat transfer to fuel pool, enclosure walls and cylinder.
10. 2 m/s cross wind blowing 45 degrees to the open door.
11. Provide various output 2-D contour slices through pool and horizontally.
12. Physical run-time five minutes (300 seconds).

The input file for this example appears in the next five pages.

```
c
c  Default Fire Model
c
c##########################################################################
*Time  nmax  ipstep  time  restart
      10000     5    0.      'W'
c  tstop  dtmax  dtmin  courant dtemp drhog  dxmu  dtwall
   300.   .02    1.E-05  1.0     2.0   1.2   1.5   1.2
*END
c##########################################################################
*EQUATIONS
c        dtemp     dtime     average
*Cafe      5        5          1
*Momentum    Type     uwm  xwm   ncor  errmx   sor   Gx    Gy  Gz
          'PRESSURE'  0.0  0.04   30   5.E-04  1.5   0.0  -9.8 0.0
*Energy   type   uwe   iter sor
        '2EQN'   0.3    9   1.5
*STABILIZE   5.000E-01  1.000E+00
*Species  name     uws  iter  sor    W    3d    1d   conserve
     5   'fuel'   0.3     9   1.5   220. 'ON'  'OFF'  'OFF'
           'o2'   0.3     9   1.5    32. 'ON'  'OFF'  'OFF'
         'soot'   0.3     9   1.5  1200. 'ON'  'OFF'  'OFF'
         'intr'   0.3     9   1.5   120. 'ON'  'OFF'  'OFF'
       'oil_pc'   0.3     9   1.5    29. 'ON'  'OFF'  'OFF'
*DIFF_REAC
  1.E-05
c       Fv_min   esoot   tinf   'view'  nvapors   Vapor-dome-Names
*SOOT   5.0E-07    1.    477.    'ON'      4   'fuel'   'o2'  'soot'
  'fuel'                'CH4'                  10.
  'intr'                'CH4'                  2.
  'soot'                'SOOT'                 5.55
  'oil_pc'              'CH2_PC'               1.
*Chemistry nreac  type       #Sp coef   spec  coef     name
             4  'Combustion'   2  1.0   'fuel'  2.87  'o2'
                              3  0.125 'soot'  3.82   'oil_pc'  .2 'intr'
                                 1e15   8000   37.7e6  300
                'Arrhenius'   1  1.0   'fuel'
                              2  0.05   'soot'  .95   'intr'
                                 2e14   42730  -3e5    350
                'Combustion'  2  1.0   'intr'  3.07    'o2'
```

```
                              1  4.07  'oil_pc'
                                 1e15  8000    42.12e6 310
                'Combustion'  2  1.0   'soot'    2.6   'o2'
                              1  3.6   'oil_pc'
                                 1e8   26500  32e6     400
*END
c###########################################################################
*REGIONS
*Fluid       Type        w   gam  pres0    name
            'igas'      29   1.4  1e5     'air'
*GasProp
            'Cond'  'Radiation'   1       .03
            'Visc'  'Constant'  2.e-5
            'Cond'  'Constant'  .035
            'Cp'    'spec_temp' 'air' 3  300   1000
                                         1580  1250
                                         2000  1350
                               'fuel' 2  300   2200
                                         2000  2200
                                'o2'  3  300   1000
                                         1580  1250
                                         2000  1350
                               'soot' 2  300   2000
                                         2000  2000
                               'intr' 2  300   2200
                                         2000  2200
                             'oil_pc' 2  300   1100
                                         2000  1500
*GasReg   #   press  temp  rux   rvy  ruz   m1   m2     m3   m4  m5
          1   1e5    300     0.    0    0   0    .233   1e-8  0  0   'air'
          2   1e5    300     0.    0    0   0    0.0   0.0    0  0   'no species'
*Structure #   Key   temp   Kg/m3    K       c      M2/m3  emis  hext tinf  alpha
           1 'CON'   300.    803.    0.      1.       0.    0.    0.   0.     1.  'Couple Cylinder     '   0 0 0 0 0
           2 'CON'   300.    880.    0.      1.       0.    0.8   0.   300.   1.  'Couple Fuel         '   0 0 0 0 0
           3 'CON'   300.    2700.   200.    300.     0.    0.2   0.   300.   1.  'Aluminum            '   0 0 0 0 0
           4 'CON'   300.    2300.   0.75312 656.90   0.    0.    0.   0.     1.  'Concrete, 1-4 dry   '   0 0 0 0 0
           5 'CON'   400.    820.    0.13000 2200.0   0.    0.3   0.   0.     1.  'JP8 Fuel            '   0 0 0 0 0
           6 'CON'   300.    7920.   14.644  502.10   0.    0.6   0.   0.     1.  'Steel, stainless 304'   0 0 0 0 0
*Heat
    1 'H'    10.0     'H=10 W/m^2K         '
```

```
*Turb  Key    Type     coef  (ax (x - x0) + by (y-y0) + cz ( z-z0) ) **d  +  e
   1 'EDDY'  'LES2'   5.0E-02  0.      0.   0.1    0.   0.      0.   0.66   1.0E-03 'Flame Default'
*Drag  Key     eps     Hdia
   1 'MOODY'  5.0E-03   1.     'Moody 1(m) dia      '
*Orfice Key  %Areax   Coefx  %Areay Coefy  %Areaz Coefz
   1 'NORMAL'  0.       1.     0.      0.     0.    0.     'X dir Blockage'
   2 'NORMAL'  0.       0.     0.      1.     0.    0.     'Y dir Blockage'
   3 'NORMAL'  0.       0.     0.      0.     0.    1.     'Z dir Blockage'
*END
c##########################################################################
*OBJECT3D    'Flame'
*MESH
c--- X dir mesh  Start  Stop  Number
'-XPWR'          -10.     -8.    3 'Domain Start'
 'XFIX'           -8.   -6.5     5 'Enclosure Start'
 'XFIX'         -6.5    -5.5     4 'CYL RAD Max'
 'XFIX'         -5.5       -5.   3 'CYL RAD Min'
 'XFIX'           -5.   -4.5     2 'CYL CENTER'
 'XFIX'         -4.5    -3.5     3 'CYL RAD Max'
'+XPWR'         -3.5       2.    8 'CYL RAD Min'
'+XPWR'            2.     10.    6 'Enclosure End'
c--- Y dir mesh
 'YFIX'         -0.5    -0.2     1 'Domain Start'
 'YFIX'         -0.2       0.    1 'CYL START'
 'YFIX'            0.    0.5     2 'BOX End'
 'YFIX'          0.5       1.    2 'CYL RAD Min'
 'YFIX'            1.    1.5     2 'CYL CEN'
 'YFIX'          1.5       3.    4 'CYL RAD Max'
 'YFIX'            3.      4.    3 'Hole End'
 'YFIX'            4.      5.    3 'Hole Start'
'+YPWR'            5.     10.    8 'Enclosure End'
c--- Z dir mesh
'-ZPWR'          -10.     -5.    7 'Domain Start'
'-ZPWR'           -5.     -2.    7 'Enclosure Start'
 'ZFIX'           -2.   -1.5     2 'Hole Start'
 'ZFIX'         -1.5       -1.   1 'CYL Rad Max'
 'ZFIX'           -1.      0.    3 'Hole Start'
 'ZFIX'            0.      1.    3 'CYL Cen'
 'ZFIX'            1.    1.5     2 'Hole end'
 'ZFIX'          1.5       2.    1 'CYL Rad Min'
```

```
'-ZPWR'            2.      5.    7 'Hole end'
'+ZPWR'            5.     10.    7 'Enclosure End'
c--------- 3-d geometry-------------
*Geom          xmin xmax  ymin ymax zmin  zmax      Gas  Wall Pwr  HT Turb  Drag  Orf
 'DOMAIN'     -10.  10.   -0.5  10. -10.   10.       1    0    0    0    1    0    0 'Air Domain'
 'ENCLOSURE'   -8.   2.   -0.5   5.  -5.   5.        1    0    0    0    1    0    0 'Thin Wall Enclosure'
 'BOX'        -10.  10.   -0.5   0. -10.   10.       1    2    0    0    0    0    0 'Couple Fuel'
 'BOX'        -10.  10.   -0.2   0. -10.   10.       1    0    0    0    1    0    0 'Opening for ground'
c cyl  axis    x0    z0   start stop    rad
 'CYL'  'Y'    -5.    0.  -0.2   0.    -1.5          1    4    0    0    1    0    0 'Ground / Pool Openin'
c cyl  axis    x0    y0   start stop    rad
 'CYL'  'Z'    -5.    1.  -1.5  1.5     0.5          1    1    0    0    0    1    0 'Calorimeter'
c             xmin xmax  ymin ymax zmin  zmax       Gas  Wall Pwr  HT Turb  Drag  Orf
 'HOLE'         2.   2     0.   3. -3.    3.         1    0    0    0    1    0    0 'Open Door'
 'HOLE'        -8.  -8     4.   5. -1.    1.         1    0    0    0    1    0    0 'Vent'
c--------- 3-d boundary conditions------------
*Boundaries   Side     Type    start stop    start stop   Arg1   Arg2  Arg3  Arg4 Arg5
             'ALL'    'CONC'     0.    0.       0.   0.     0.   0.233  0.    0.    0.
             'ALL'    'FLUX'     0.    0.       0.   0.     0.    300.
             'ALL'    'WIND'     0.    0.       0.   0.    -1.44    0.      -1.4400
             'TOP'    'HYDRO'    0.    0.       0.   0.     300.
*END
c##########################################################################
c     hydrostatic pressure initial condition
*INIT
        'hydro'
*end
c##########################################################################
c define all 1-d subgrids
*OBJECT1D  'Calorimeter'
*1DAUTO
    1   10    1    6    0    0    0    0    0  1.0E-02 'SS Wall'
*END
c##########################################################################
*OBJECT1D  'Thin Walls'
*1DAUTO
    1    4    1    6    0    0    0    0    0  2.0E-03 'wall'
*END
c##########################################################################
*OBJECT1D  'Pool'
```

```
*1DAUTO
    1   10    1    5    0    0    0    0    0  0.1     'fuel'
*BOUNDARY
c   side   Type   Tboil  Hfg     fuel Cp   Ev_0   time0  Ev_mx Ev_mn  Mass-Frac
   'Left' 'EVAP'   450. 3.0E+05  2200.0   7.0E-02   3.    0.2   .01   1. 0. 0. 0. 0.
*END
c##########################################################################
c     Coupling Between 3-d and 1-d subgrids
 *COUPLING  'AUTO'
#surf type   name    search  couple   xi  xf   yi   yf   zi  zf
    1  'Q'  'FLAME'   'BOX'  'G-S2' -10. 10.  -0.5  10. -10. 10.
            'Pool'    '1DA'  'none'   1
#surf type   name    search  couple   xi  xf   yi   yf   zi  zf
    2 'Q'   'FLAME'   'BOX'  'G-S1' -10. 10.  -0.5  10. -10. 10.
        'Calorimeter' '1DA'  'none'   1
#surf type   name    search  couple                xi  xf   yi   yf   zi  zf
    3 'Q'   'FLAME'   'BOX' 'Thin Wall Enclosure' -10. 10. -0.5 10. -10.  10.
         'Thin Walls' '1DA'  1 1000
*END
c##########################################################################
*OUTPUT
*OFF
 1 'TS'                   0.  0.  0.  0.  0.  0.
*GMV 'ascii' delay dt   xi xf yi yf zi zf  n0
               0.  1.    0  0  0  0  0  0  -1
*BINARY-2D  Number Tstart  dt
              4      0.     1.
c Binary-2D  filename    axis  start stop start stop  plane
          'Center.bin'   'Z'     0.    0.   0.    0.   0.     'none'
          'YSlice.bin'   'Y'     0.    0.   0.    0.   1.     'none'
         'Ceiling.bin'   'Y'     0.    0.   0.    0.   4.9    'none'
          'Xslice.bin'   'X'     0.    0.   0.    0.  -5.     'XSpec.bin'
*FIRE
*Grid
*END
```

## 11.2 Example Problem Results

Figures 9 shows a front view of the model with an open door and ground surface. This figure depicts the domain and enclosure size. The walls of the enclosure are colored by local temperature after a two-minute fire.

These images were created by the Geomview code (on UNIX systems). On PCs, similar images can be created by any VRML viewer the user has installed.

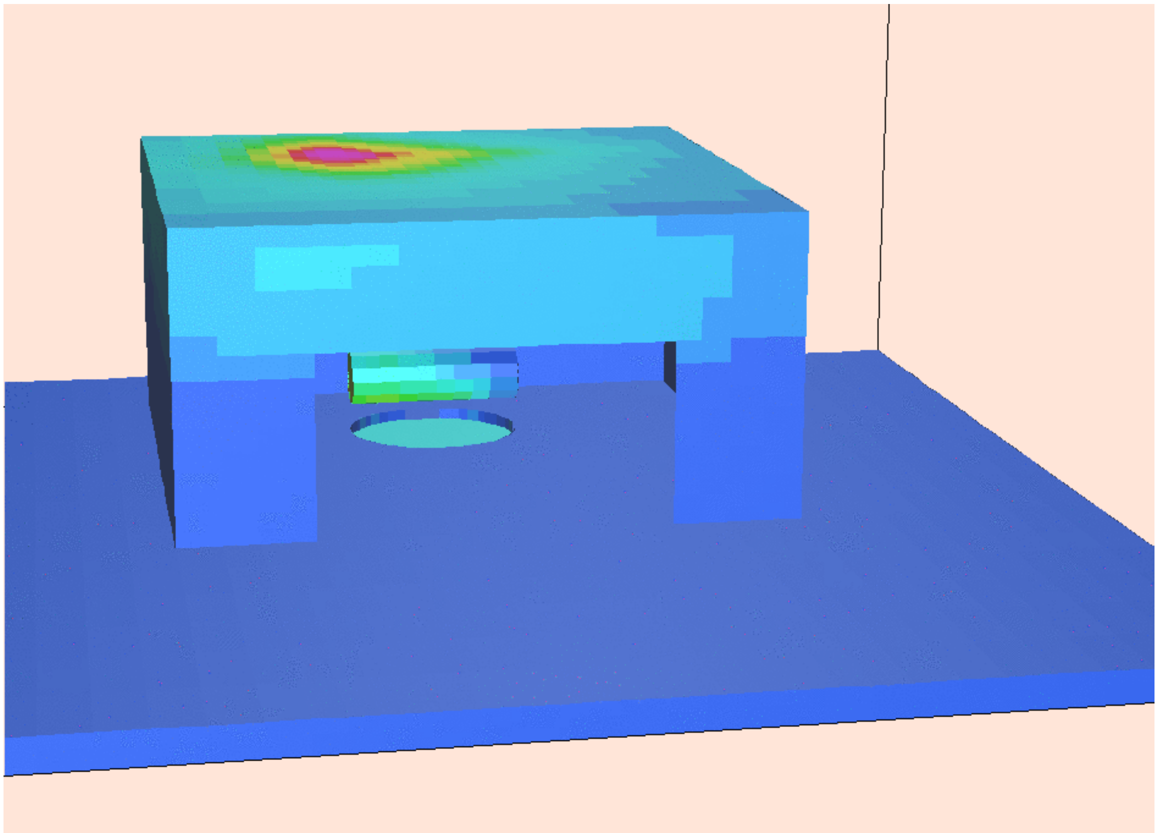

**Figure 9. A frontal view of the example problem. Shown are the enclosure, calorimeter, fuel pool, and open door. Colors represent temperature after two-minute fire.**

The exterior cross wind in Figure 9 blows from left to right. This crosswind is captured by the open door and causes a circular flow in the enclosure counter clockwise viewed from above (i.e. vorticity vector points upward). The temperatures depicted in the figure shows the fire tilted to the left due to the air circulation.

Figure 10 is a close up of the 1-centimeter thick cylindrical calorimeter located above the pool. The fuel pool is shown to be depressed below ground level. The walls of the calorimeter are colored by local temperature after a two-minute fire. As shown, they are significantly cooler than the 2-millimeter thick enclosure wall hot spot shown in the previous figure.

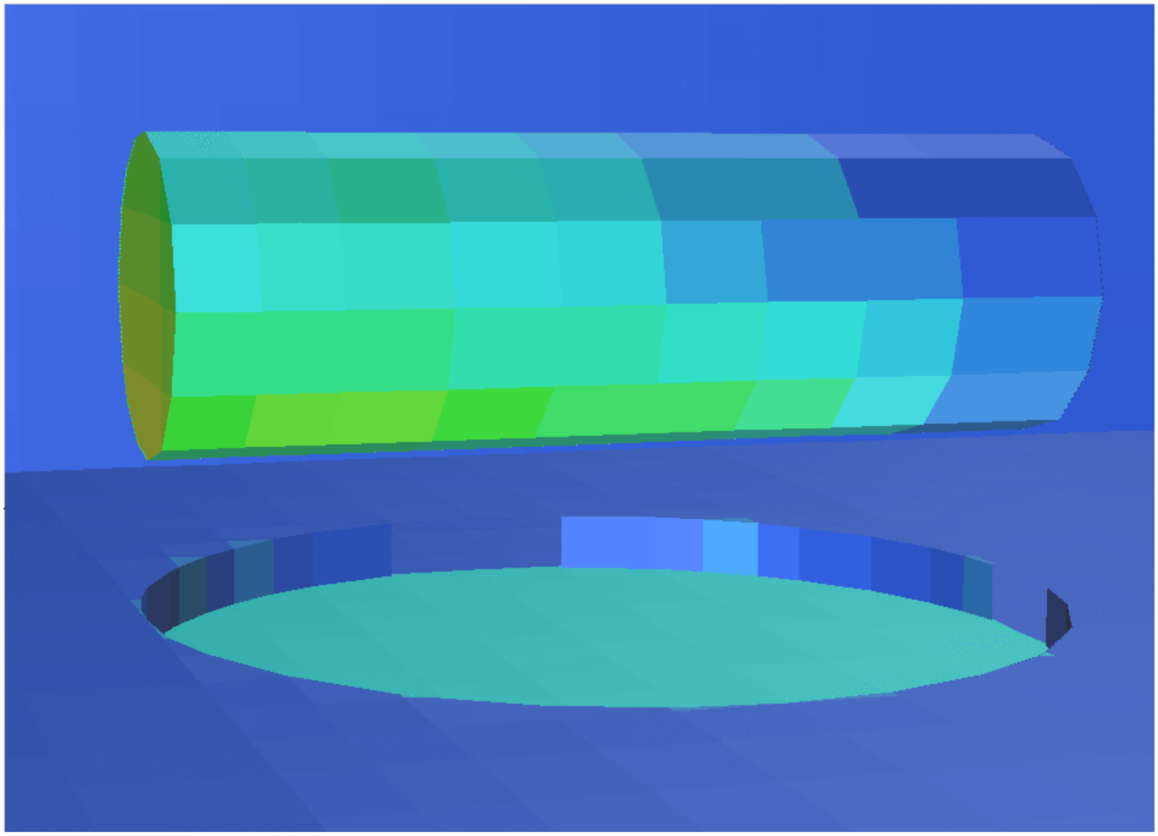

**Figure 10. A close up view of the 1-centimeter thick calorimeter. The colors represent the local temperature after a two-minute fire.**

The colored squares not only represent the local temperature, but also show the size and orientation of the 1-D subgrids. These 1-D subgrids are used for local grid refinement and allow the temperature of the 1 centimeter thick wall to be accurately calculated in time. The 1-D subgrid mesh size is 1 millimeter, compared to the 3-D mesh size, which is approximately 20 centimeter. This results in a factor of 200 in grid refinement.

Figure 11 is an isosurface image of 0.5 ppm soot volume fraction colored by the local gas temperature. It depicts the fire and smoke shape. The enclosure has been cut away to show the interior.

The isosurface image was created by the GMV code on a UNIX system. On PCs, GMV can be downloaded and installed but requires an X server (for example, from Cygwin). Alternatively, the user could purchase Tecplot, Plot 3d, or Fieldview to create similar isosurface images.

The tilt of the fire is quite evident in the image. The tilt of the fire is caused by the circulation of the air in the room, which in turn was caused by the exterior crosswind.

Also depicted in the figure is the stratified smoke layer in the room.

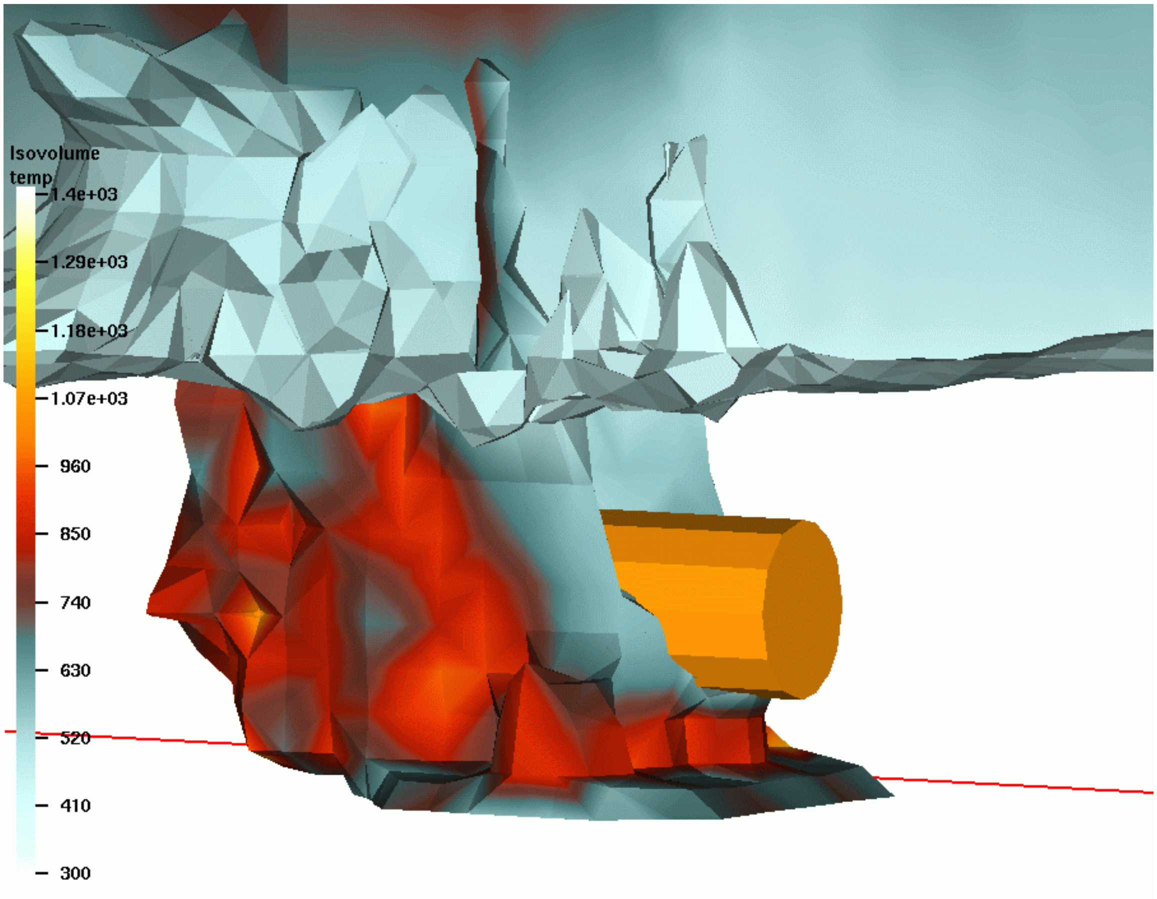

**Figure 11. An isosurface image of 0.5 ppm soot volume fraction colored by the local gas temperature.**

This page intentionally left blank.

# 12. References

Anderson, Tannehill, Pletcher, *Computational Fluid Mechanics and Heat Transfer,* McGraw Hill, 1984.

Bird, R.B., W.E. Stewart, and E.N. Lightfoot, *Transport Phenomena*, Wiley New York (1960).

Choi, M.Y., G.W. Mulholland, A. Hamins, and T. Kashwagi, "Comparisons of the Soot Volume Fraction using Gravimetric and Light Extinction Techniques." Combustion and Flame V102, pp 161-169, 1995.

Ferzinger, J.H., 'Subgrid-Scale Modeling' In *Large Eddy Simulations of Complex Engineering and Geophysical Flows*, Eds. B. Galperin and S.A. Orszag, Cambridge University Press, Cambridge, pp. 37-54, 1993.

Greiner, M., and A. Suo-Anttila, "Validation of the Isis Computer Code for Simulating Large Pool Fires Under a Varity of Wind Conditions," *ASME J. Pressure Vessel Technology,* Vol. 126, pp. 360-368, 2004. Journal Paper, (Also presented at the *ASME Pressure Vessels and Piping Conference*, Cleveland, Ohio, July 20 - 24, 2003 Conference Paper).

Gritzo, L.A., J.A. Moya, and D. Murray, "Fire characterization and object thermal response for a large flat plate adjacent to a large JP4 fuel fire," Sandia National Laboratories Report SAND97-0047, p. 22, January 1997.

Gritzo, L.A., Y.R. Sivathaunu, and W. Gill, "Transient Measurements of Radiative Properties, Soot Volume Fraction, and Soot Temperature in a Large Pool Fire," *Combustion Science and Technology,* Vol. 139, pp. 113-136, 1998.

Hamis, A., J. Fischer, T. Kashiwag, M.E. Klassen, and J.P. Gore, "Heat Feedback to the Fuel Surface in Pool Fires," *Combustion Science and Technology*, Vol. 97, No 1-3, pp. 36-62, 1994.

Hirt, C.W. and J.M. Sicilian, "A Porosity Technique for the Definition of Obstacles in Rectangular Cell Meshes," 4th International Conference on Ship Hydrodynamics Washington, DC, September 1985.

Issa, R.I., "Solution of the Implicitly Discretised Fluid Flow Equations by Operator Splitting," *Journal Computational Physics*, 62: 40-65, 1985.

Ju, H., M. Greiner, and A. Suo-Anttila, "Computer Simulations of a Generic Truck Cask in a Regulatory Fire Using the Container Analysis Fire Environment (CAFE) Code," *Int. Journal of Radioactive Materials Transport*, Vol. 13, pp. 35-40, 2002. Journal Paper. (Also presented at the *13th International Symposium on the Packaging and Transportation of Radioactive Material (PATRAM)*, CD-ROM, Chicago, Illinois, September 3-7, 2001.).

Lopez, C., J. Koski, and A. Suo-Anttila, "Development and Use of the CAFE-3D Code for Analysis of Radioactive Material Packages in Fire Environments," *Institute of Nuclear Materials Management 44h Annual Meeting*, September 2003.

Lopez, C., A. Suo-Anttila, J. Koski, and I. Khalil, "Calibration of the CAFE-3D Code with Controlled Indoor Fire Data," Proceedings International Symposium on the Packaging and Transportation of Radioactive Materials (PATRAM), 2004.

Magnussen, B.F. and B.H. Hjertager, "On Mathematical Modeling of Turbulent Combustion with Special Emphasis on Soot Formation and Combustion," *16th International Symposium on Combustion*, *The Combustion Institute*, Pittsburg Pa. PP 719-729, 1976.

McDonough, J.M. *Introductory Lectures on Turbulence*. Dept. of Mechanical Engineering and Mathematics, University of Kentucky, 2004.

Modest, M.F., *Radiative Heat Transfer*, McGraw Hill Inc., 1993.

Murphy, J.J. and C.R. Shaddix. "Soot Properties and Species Measurements in a Two-Meter Diameter JP-8 Pool Fire," SAND2003-8246, Sandia National Laboratories, July, 2003.

Murphy, J.J. and C.R. Shaddix, "Soot Properties and Species Measurements in a Two-Meter Diameter JP-8 Pool Fire: 2003 Test Series," SAND2004-8085, Sandia National Laboratories, March 2004.

Patankar S.V., and D.B. Spalding, "A Calculation Procedure for Heat, Mass, and Momentum Transfer in Three Dimensional Parabolic Flows," *International Journal Heat and Mass Transfer*. vol. 15, pp. 1787-1806, 1972.

Pomraning, E. and C.J. Rutland, 2002, "A Dynamic One-Equation Non-Viscosity LES Model," *AIAA Journal*, Vol. 40, No. 4, pg. 689-701.

Siegel, R. and J.R. Howell, *Thermal Radiation Heat Transfer, 2nd Edition*. McGraw-Hill Book Company, 1981.

Smagorinsky, J., "General Circulation Experiments with the Primitive Equations," *Monthly Weather Review*, Vol. 91, pp. 99-165, 1963.

Said, R., A. Garo, and R. Borghi, "Soot Formation Modeling for Turbulent Flames," *Combustion and Flame*," Vol. 108, pp. 71-86, 1997.

Society Fire Protection Engineers, *Fire Protection Engineering, 2nd Edition*, National Fire Protection Association Publication, June 1995.

Suo-Anttila, A., "F2D Users Manual: A Two-Dimensional Compressible Gas Flow Code," Sandia National Laboratories Report, SAND92-7343, June 1993.

Suo-Anttila, A.J., J.A. Koski, and L.A. Gritzo, "CAFE: A Computer Tool for Accurate Simulation of the Regulatory Pool Fire Environment for Type B Packages," SAND99-0625C, March 1999.

Suo-Anttila, A.J., M. Greiner, J. Dunham, A. Zivkovich, and K.C. Wagner. "Isis-3D Software Validation Report," ITSC-Isis-SVR-001, March 2000.

Suo-Anttila, A., CAFE 2D Users Manual. Letter report to J.A. Koski, Sandia National Laboratories, October 2000.

Suo-Anttila, A., K.C. Wagner, and M. Greiner, "Analysis of Enclosure Fires Using the Isis-3D$^{TM}$ CFD Engineering Analysis Code," *Proceedings of ICONE12, 12th International Conference on Nuclear Engineering*, Arlington, Virginia USA, April 25-29, 2004.

Suo-Anttila J.M. and L.A. Gritzo, "Thermal Measurements from a Series of Tests with a Large Cylindrical Calorimeter on the Leeward Edge of a JP-8 Pool Fire in Crossflow," Sandia National Laboratories Report, SAND2001-1986, June 2001.

U.S. Nuclear Regulatory Commission, "Packaging and Transportation of Radioactive Material," Rules and Regulations, Title 10, Part 71, *Code of Federal Regulations*, 2000.

## DISTRIBUTION:

1 Nancy Slater-Thompson
Office of Civilian Radioactive Waste
Management
RW-31E /Forrestal Building
U.S. Department of Energy
1000 Independence Ave., S.W.
Washington, DC 20585

6 U.S. Nuclear Regulatory Commission
NMSS-Spent Fuel Project Office
Attn.: Christopher S. Bajwa (3)
Antonio F. Dias (3)
Mail Stop O-13D13
11555 Rockville Pike
Rockville, MD 20852

3 University of Nevada, Reno Campus
Mechanical Engineering Department
Attn.: Prof. Miles Greiner
MS 312
Reno, Nevada 89557

3 U.S. Department of Transportation
Federal Railroad Administration
Attn.: Kevin R. Blackwell
RRS-12, Mail Stop 25
1120 Vermont Ave. N.W.
Washington, DC 20590

1 Gesellschaft für Anlagen-
und Reaktorsicherheit (GRS) mbH
Attn.: Wenzel Brücher
Schwertnergasse 1
50667 Cologne, Germany

1 Jorman A. Koski
5624 Moon St. NE
Albuquerque, NM 87111

| | | |
|---|---|---|
| 1 | MS 0701 | P.B. Davies, 6100 |
| 1 | MS 0718 | D.R. Miller, 6141 |
| 1 | MS 0718 | K.B. Sorenson, 6142 |
| 1 | MS 0718 | J.J. Danneels, 6143 |
| 2 | MS 1146 | A.J. Suo-Anttila, 6784 |
| 2 | MS 0718 | I. Khalil, 6141 |
| 5 | MS 0718 | C. Lopez, 6141 |
| 1 | MS 0718 | D.J. Ammerman, 6141 |
| 1 | MS 0719 | S.W. Webb, 6141 |
| 1 | MS 1135 | A.L. Brown, 9132 |
| 1 | MS 9018 | Central Technical Files, 8945-1 |
| 2 | MS 0899 | Technical Library, 9616 |
| 1 | MS 0161 | Patent and Licensing Office, 11500 |