U.S. DEPARTMENT OF COMMERCE

TECHNOLOGY ADMINISTRATION

National Institute of Standards and Technology

NIST Technical Note 1299

CFAST, the Consolidated Model of Fire Growth and Smoke Transport

Richard D. Peacock
Glenn P. Forney
Paul A. Reneke
Rebecca M. Portier
Walter W. Jones

Contents

Figures and Tables

Executive Summary

Analytical models for predicting fire behavior have been evolving since the late 1970's. Individuals have tried to describe in mathematical language the various phenomena which have been observed in fire growth and spread. These separate representations often describe only a small part of a fire experience. When combined, these separate pieces interact and form a complex computer code intended to give an estimate of the expected course of a fire based upon given input parameters. These complete analytical models have progressed to the point of providing predictions of fire behavior with an accuracy suitable for most engineering applications.

CFAST is a member of a class of models referred to as zone or finite element models. This means that each room is divided into a small number of volumes (called zones), each of which is assumed to be internally uniform. That is, the temperature and smoke and gas concentrations within each zone are *assumed* to be exactly the same at every point. In CFAST, each room is divided into two layers. Since these layers represent the upper and lower parts of the room, conditions within a room can only vary from floor to ceiling, and not horizontally. This assumption is based on experimental observations that in a fire, room conditions do stratify into two distinct layers. While we can measure variations in conditions within a layer, these are generally small compared to differences between the layers. This assumption places some limitations on the predictions which such a model can make. As the modeling evolves, many of these assumptions are being lifted.

CFAST is based on solving a set of equations that predict the change in the enthalpy and mass over time. The equations are derived from the conservation equations for energy, mass, momentum, and an equation of state, in this case, the ideal gas law. The conservation equations are fundamental to physical systems, and must hold in all cases. These equations are rearranged to form a set of predictive equations for the sensible variables in each compartment. CFAST is formulated as a set of ordinary differential equations. It was the first model of fire growth and smoke spread to cast the entire model in this form and was done because of the efficiency of solving the conservation equations this way.

The accuracy of the current model is limited by the fire being uninfluenced by radiation from its surroundings, and by our inability to quantify accurately the effects of fire on people and their actions. Research is underway to better understand radiation enhanced burning and predict fire growth and spread, fuel mass loss rate and combustion product generation rates under those conditions. Otherwise, CFAST provides reasonable predictions of the several experiments examined in this paper. Although differences between the model and the experiments exist, they can be explained by limitations of the model and of the understanding of experimental data.

As with any theoretical model, there are pieces which have been omitted and others which could be implemented more completely. However, with an understanding of the relative weaknesses and strengths of both the model and of the experiments used to verify the model, the user can develop confidence in using such models for a wide range of simulations.

This document is intended to provide a sufficiently detailed explanation of the basis and structure of the model that someone could modify CFAST and add a new phenomenon.

Nomenclature

A_d duct surface area (m^2)
A_o area of the inlet, outlet, duct, contraction, or expansion joint, coil, damper, bend, filter, and so on in a mechanical ventilation system. (m^2)
A_{room} floor area of a room (m^2)
A_{slab} cross-sectional area for horizontal flow (m^2)
A_v area of ceiling or floor vent (m^2)
A_w wall surface area (m^2)
b_i coefficients for adsorption and desorption of HCl
C flow coefficient for horizontal flow of gas through a vertical vent
C_{LOL} Lower oxygen limit coefficient, the fractional burning rate constrained by available oxygen, eq (39)
C_o characterisitc flow coefficient
C_w wind coefficient – dot product of the wind vector and vent direction
CO/CO_2 ratio of the mass of carbon monoxide to the mass of carbon dioxide in the pyrolysis of the fuel
CO_2/C ratio of the mass of carbon dioxide to the mass of carbon in the pyrolysis of the fuel
c_k heat sources for the k'th wall segment (W)
c_p heat capacity of air at constant pressure (J/kg K)
c_v heat capacity of air at constant volume (J/kg K)
D effective diameter of ceiling or floor vent (m)
D_e effective duct diameter (m)
d_{HCl} rate of deposition of Hcl onto a wall surface, eq. (84) (kg/s)
E_i internal energy in layer i (W)
F friction factor
F_{k-j} configuration factor
g gravitational constant (m^3/s)
G conductance
Gr Grashof number
H_c heat of combustion of the fuel (J/kg)
h_c convective heat transfer coefficient (J/m^2 K)
$\dot{h}_i$ rate of addition of enthalpy into layer i (W)
h_l convective heat transfer coefficient in ceiling boundary layer (J/m^2 K)
$\tilde{h}$ characteristic convective heat transfer coefficient
H height of the ceiling above a fire source (m)
H/C ratio of the mass of hydrogen to the mass of carbon in the pyrolysis of the fuel
H_2O/H ratio of the mass of water to the mass of hydrogen in the pyrolysis of the fuel
HCl/C ratio of the mass of hydrogen chloride to the mass of carbon in the pyrolysis of the fuel
HCl/f ratio of the mass of hydrogen chloride to the total mass of the fuel
HCN/C the mass of hydrogen cyanide to the mass of carbon in the pyrolysis of the fuel
HCN/f ratio of the mass of hydrogen cyanide to the total mass of the fuel
k mass transfer coefficients for HCl deposition
l characteristic length for convective heat transfer (m)
m_i total mass in layer i (kg)
$m_{i,j}$ mass flow from node i to node j in a mechanical ventilation system (kg/s)
$\dot{m}_b$ burning rate of the fuel (perhaps constrained by available oxygen) (kg/s)
$\dot{m}_c$ production rate of carbon during combustion (kg/s)

m_d	mass flow in duct (kg/s)
$\dot{m}_e$	rate of entrainment of air into the fire plume (kg/s)
$\dot{m}_f$	pyrolysis rate of the fuel (before being constrained by available oxygen) (kg/s)
$\dot{m}_i$	rate of addition of mass into layer i (kg/s)
O/C	ratio of the mass of oxygen to the mass of carbon in the pyrolysis of the fuel
P	pressure (Pa)
P_{ref}	reference pressure (Pa)
Pr	Prandtl number
Q_c	total convective heat transfer (W)
Q_{eq}	dimensionless plume strength at layer interface
Q_f	total heat release rate of the fire (W)
Q_H	dimensionless plume strength at the ceiling
Q_r	total readiative heat transfer (W)
r	radial distance from point source fire (m)
R	universal gas constant (J/kg K)
R_e	Reynolds number
S	vent shape factor for vertical flow
S/C	ratio of the mass of soot to the mass of carbon in the pyrolysis of the fuel
t	time (s)
T_{amb}	ambient temperature (K)
T_d	duct temperature (K)
T_e	temperature of gas entrainment into the fire plume (K)
T_g	gas temperature (K)
T_i	temperature of layer i (K)
T_{in}	duct inlet temperature (K)
T_k	temperature of the k'th wall segment (K)
T_{out}	duct outlet temperature (K)
T_p	temperature of the plume as it intersects the upper layer (K)
T_w	wall temperature (K)
v	gas velocity (m/s)
V_d	duct volume (m^3)
V_i	volume of layer i (m^3)
Y	mass fraction of a species in a layer
Y_{LOL}	lower oxygen limit for oxygen constrained burning, expressed as a mass fraction
z	height over which entrainment takes place (m)
Z	height (m)

α	absorption coefficient of the gas (m^{-1})
ΔP	pressure offset from reference pressure, $P - P_{ref}$ (Pa)
γ	ratio of c_p/c_v
ϵ_k	emissivity of the k'th wall segment
κ	thermal conductivity (J/m s K)
ν	kinematic viscosity (m^2/s)
ρ_d	density of gas in a duct (kg/m^3)
ρ_i	density of gas in layer i (kg/m^3)
σ	Stefan-Boltzman constant (5.67×10^{-8} W/m^2K^4)
τ	transmissivity factor
χ_c	fraction of the heat release rate of the fire which goes into convection
χ_r	fraction of the heat release rate of the fire which goes into radiation

CFAST, the Consolidated Model of Fire Growth and Smoke Transport

Richard D. Peacock, Glenn P. Forney, Paul Reneke, Rebecca Portier, and Walter W. Jones

Building and Fire Research Laboratory
National Institute of Standards and Technology

Abstract

CFAST is a zone model capable of predicting the environment in a multi-compartment structure subjected to a fire. It calculates the time evolving distribution of smoke and fire gases and the temperature throughout a building during a user-specified fire. This report describes the equations which constitute the model, the physical basis for these equations, data which are used by the model, and details of the operation of the computer program implementing the model. The means by which one can add new phenomena are detailed, as are the variables and structure of the model.

A set of comparisons between the model and a range of real-scale fire experiments is presented. In general, the CFAST model compares favorably with the experiments examined in this paper. Although differences between the model and the experiments were clear, they can be explained by limitations of the model and of the experiments.

1 Introduction and Overview

Analytical models for predicting fire behavior have been evolving since the 1960's. Over the past two decades, the completeness of the models has grown. In the beginning, the focus of these efforts was to describe in mathematical language the various phenomena which were observed in fire growth and spread. These separate representations have typically described only a small part of a fire. When combined though, they can create a complex computer code intended to give an estimate of the expected course of a fire based upon given input parameters. These analytical models have progressed to the point of providing predictions of fire behavior with an accuracy suitable for most engineering applications. In a recent international survey [1], 36 actively supported models were identified. Of these, 20 predict the fire generated environment (mainly temperature) and 19 predict smoke movement in some way. Six calculate fire growth rate, nine predict fire endurance, four address detector or sprinkler response, and two calculate evacuation times. The computer models now available vary considerably in scope, complexity, and purpose. Simple “room filling” models such as the Available Safe Egress Time (ASET) model [2] run quickly on almost any computer, and provide good estimates of a few parameters of interest for a fire in a single compartment. A special purpose model can provide a single function. For example, COMPF2 [3] calculates post-flashover room temperatures and LAVENT [4] includes the interaction of ceiling jets with fusible links in a room containing ceiling vents and draft curtains. Very detailed models like the HARVARD 5 code [5] or FIRST [6] predict the burning behavior of multiple items in a room, along with the time-dependent conditions therein.

In addition to the single-room models mentioned above, there are a smaller number of multi-room models which have been developed. These include the BRI transport model [7], the HARVARD 6 code [8] (which is a multi-room version of HARVARD 5), FAST [9], CCFM [10] and the CFAST model discussed below [11].

Although the papers are several years old, Mitler [12] and Jones [13] reviewed the underlying physics in several of the fire models in detail. The models fall into two categories: those that start with the principles of conservation of mass, momentum, and energy; the other type typically are curve fits to particular experiments or series of experiments, used in order to try and tease out the underlying relationship among some parameters. In both cases, errors arise in those instances where a mathematical short cut was taken, a simplifying assumption was made, or something important was not well enough understood to include.

Once a mathematical representation of the underlying science has been developed, the conservation equations can be re-cast into predictive equations for temperature, smoke and gas concentration, and other parameters of interest, and are coded into a computer for solution. The environment in a fire is constantly changing. Thus the equations are usually in the form of *differential equations*. A complete set of equations can compute the conditions produced by the fire at a given time in a specified volume of air. Referred to as a *control volume*, the model assumes that the predicted conditions within this volume are uniform at any time. Thus, the control volume has one temperature, smoke density, gas concentration, etc.

Different models divide the building into different numbers of control volumes depending on the desired level of detail. The most common fire model, known as a *zone model*, generally uses two control volumes to describe a room – an upper layer and a lower layer. In the room with the fire, additional control volumes for the fire plume or the ceiling jet may be included to improve the accuracy of the prediction (see Figure 1).

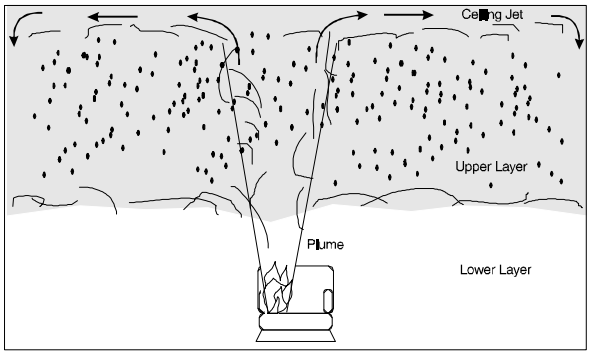


Figure 1. Zone Model Terms.

This two-layer approach has evolved from observation of such layering in real-scale fire experiments. Hot gases collect at the ceiling and fill the room from the top. While these experiments show some variation in conditions within the layer, these are small compared to the differences between the layers. Thus, the zone model can produce a fairly realistic simulation under most conditions.

Other types of models include *network models* and *field models*. The former use one element per room and are used to predict conditions in spaces far removed from the fire room, where temperatures are near ambient and layering does not occur. The field model goes to the other extreme, dividing the room into thousands or even hundreds of thousands of grid points. Such models can predict the variation in conditions within the layers, but typically require far longer run times than zone models. Thus, they are used when highly detailed calculations are essential.

CFAST is a zone model used to calculate the evolving distribution of smoke and fire gases and the temperature throughout a building during a fire. CFAST is the result of a merger of ideas that came out of the FAST [9] and the CCFM.VENTS [10] development projects. The organization of the CFAST suite of programs is thus a combination of the two models. The primary focus of this paper is to lay out the details of CFAST, including the equations used and the assumptions made. This is done with a view toward modification of the computer code, and the variables which are used are discussed and made explicit. In addition, the structure of the model is shown. Although it may not be all inclusive, CFAST has demonstrated the ability to make reasonably good predictions. Also, it has been subject to close scrutiny to insure its

correctness. Thus it forms a prototype for what constitutes a reasonable approach to modeling fire growth and the spread of smoke and toxic gases.

With this level of detail, researchers not intimately involved in the development of CFAST should be able to add to the model in a straightforward manner. Independent or cooperative efforts to enhance the capabilites of the model are encouraged. Model developers can use this version of the model for open or proprietary additions to the model or as the basis for new models. While encouraging additions to the model, the role of the National Institute of Standards and Technology (NIST) in the development must be clearly defined. NIST will continue to develop and document algorithms and the model in total. Depending on the extent others validate and document their additions to the model in the refereed literature, NIST may include them in its future versions. Thus, non-developers will continue to have access to an increasingly capable model from NIST, and perhaps from others as well.

This report provides an overview of the assumptions and limitations of CFAST as well as an in-depth presentation of the fundamental equations and the way in which they are implemented in the code. Values of internal constants and coefficients, and references to the literature sources from which these all come are included. A companion report [14] provides details of the use of the model.

1.1 The Model

CFAST is a member of a class of models referred to as zone or finite element models. This means that each room is divided into a small number of volumes (called layers), each of which is assumed to be internally uniform. That is, the temperature, smoke and gas concentrations within each layer are *assumed* to be exactly the same at every point. In CFAST, each room is divided into two layers. Since these layers represent the upper and lower parts of the room, conditions within a room can only vary from floor to ceiling, and not horizontally. This assumption is based on experimental observations that in a fire, room conditions do stratify into two distinct layers. While we can measure variations in conditions within a layer, these are generally small compared to differences between the layers.

CFAST is based on solving a set of equations that predict state variables (pressure, temperature and so on) based on the enthalpy and mass flux over small increments of time. These equations are derived from the conservation equations for energy mass, and momentum, and the ideal gas law. These conservation equations are always correct, everywhere. Thus any errors which might be made by the model cannot come from these equations, but rather come from simplifying assumptions or from processes left out because we don't know how to include them. Examples of each source of error will be highlighted in the following discussion.

1.2 Fires

Within CFAST, a fire is a source of fuel which is released at a specified rate. This fuel is converted into enthalpy (the conversion factor is the heat of combustion) and mass (the conversion factor is the yield of a particular species) as it burns. Burning can take place in the portion of the plume in the lower layer (if any), in the upper layer, or in a door jet. For an unconstrained fire, the burning will all take place within the fire plume. For a constrained fire, burning will take place where there is sufficient oxygen. Where insufficient oxygen is entrained into the fire plume, unburned fuel will successively move into and burn in: the upper layer

of the fire room, the plume in the doorway to the next room, the upper layer of the next room, the plume in the doorway to the third room, and so forth until it is consumed or gets to the outside.

This version of CFAST includes the ability to track, independently, multiple fires in one or more rooms of the building. These fires are treated as totally separate entities, i.e., with no interaction of the plumes or radiative exchange between fires in a room.

Like all current zone fire models, this version of CFAST does not include a pyrolysis model to predict fire growth. Rather pyrolysis rates for each fire modeled define the fire history. The similarity of that input to the real fire problem of interest will determine the accuracy of the resulting calculation. The user must account for any interactions between the fire and the pyrolysis rate. Planned future research should remove this limitation.

1.3 Plumes and Layers

Above any burning object, a plume is formed which is not considered to be a part of either layer, but which acts as a pump for enthalpy and mass from the lower layer into the upper layer (upward only). For the fire plume, CFAST does not use a point source approximation, but rather uses an empirical correlation to determine the amount of mass moved between layers by the plume.

Two sources exist for moving enthalpy and mass between the layers within and between rooms. Within the room, the fire plume provides one source. The other source of mixing between the layers occurs at vents such as doors or windows. Here, there is mixing at the boundary of the opposing flows moving into and out of the room. The degree of mixing is based on an empirically-derived mixing relation. Both the outflow and inflow entrain air from the surrounding layers. The flow at vents is also modeled as a plume (called the door plume or jet), and uses the same equations as the fire plume, with two differences. First, an offset is calculated to account for entrainment within the doorway and second, the equations are modified to account for the rectangular geometry of vents compared to the round geometry of fire plumes. All plumes within the simulation entrain air from their surroundings according to an empirically-derived entrainment relation. Entrainment of relatively cool, non-smoke laden air adds oxygen to the plume and allows burning of the fuel. It also causes it to expand as the plume moves upward in the shape of an inverted cone. The entrainment in a vent is caused by bi-directional flow and results from a phenomenon called the Kelvin-Helmholz instability. It is not exactly the same as a normal plume, so some error arises when this entrainment is approximated by a normal plume entrainment algorithm.

While experiments show that there is very little mixing between the layers at their interface, sources of convection such as radiators or diffusers of heating and air conditioning systems, and the downward flows of gases caused by cooling at walls, will cause such mixing. These are examples of phenomena which are not included because the theories are still under development. Also, the plumes are *assumed* not to be affected by other flows which may occur. For example, if the burning object is near the door the strong inflow of air will cause the plume axis to lean away from the door and affect entrainment of gases into the plume. Such effects are not included in the model.

As discussed above, each room is divided into two layers, the upper and lower. At the start of the simulation, the layers in each room are initialized at ambient conditions and by default, the upper layer volume set to 0.001 of the room volume (an arbitrary, small value set to avoid the potential mathematical problems

associated with dividing by zero). Other values can be set. As enthalpy and mass are pumped into the upper layer by the fire plume, the upper layer expands in volume causing the lower layer to decrease in volume and the interface to move downward. If the door to the next room has a soffit, there can be no flow through the vent from the upper layer until the interface reaches the bottom of that soffit. Thus in the early stages the expanding upper layer will push down on the lower layer air and force it into the next compartment through the vent by expansion.

Once the interface reaches the soffit level, a door plume forms and flow from the fire room to the next room is initiated. As smoke flow from the fire room fills the second room, the lower layer of air in the second room is pushed down. As a result, some of this air flows into the fire room through the lower part of the connecting doorway (or vent). Thus, a vent between the fire room and connecting rooms can have simultaneous, opposing flows of air. All flows are driven by pressure differences and density differences that result from temperature differences and layer depths. Thus the key to getting the right flows is to correctly distribute the fire's mass and enthalpy between the layers.

1.4 Vent Flow

Flow through vents is the dominant phenomenon in a fire model because it fluctuates most rapidly and transfers the greatest amount of enthalpy on an instantaneous basis of all the source terms. Also, it is most sensitive to changes in the environment. Flow through vents comes in two varieties. The first we refer to as horizontal flow. It is the flow which is normally thought of in discussing fires. It encompasses flow through doors, windows and so on. The other is vertical flow and can occur if there is a hole in the ceiling or floor of a compartment. This latter phenomena is particularly important in two disparate situations: a ship, and the role of fire fighters doing roof venting.

Flow through normal vents is governed by the pressure difference across a vent. There are two situations which give rise to flow through vents. The first, and usually thought of in fire problems, is that of air or smoke which is driven from a compartment by buoyancy. The second type of flow is due to expansion which is particularly important when conditions in the fire environment are changing rapidly. Rather than depending entirely on density differences between the two gases, the flow is forced by volumetric expansion. The earlier version of this model did not solve this part of the problem entirely correctly. In most cases the differences are small except for rapidly changing situations. However, these small differences become very important if we wish to follow flows due to small pressure differences, such as will occur in a mechanical ventilation system. Atmospheric pressure is about 100 000 Pa, fires produce pressure changes from 1 to 1000 Pa and mechanical ventilation systems typically involve pressure differentials of about 1 to 100 Pa. In order to solve these interactions correctly, we must be able to follow pressure differences of $\approx$ 0.1 Pa out of 100 000 Pa for the overall problem, or 10^{-4} for adjacent compartments.

1.5 Heat Transfer

Heat transfer is the mechanism by which the gas layers exchange energy with their surroundings. Convective transfer occurs from the layers to the room surfaces. The enthalpy thus transferred in the simulations conducts through the wall, ceiling, or floor in the direction perpendicular to the surface only. CFAST is more advanced than most models in this field since it allows different material properties to be used for the ceiling, floor, and walls of each room (although all the walls of a room must be the same). Additionally, CFAST

uniquely allows each surface to be composed of up to three distinct layers for each surface, which are treated separately in the conduction calculation. This not only produces more accurate results, but allows the user to deal naturally with the actual building construction. Material thermophysical properties are *assumed* to be constant, although we know that they actually vary with temperature. This assumption is made because data over the required temperature range is scarce even for common materials, and because the variation is relatively small for most materials. However the user should recognize that some materials may change mechanical properties with temperature. These effects are not modeled.

Radiative transfer occurs among the fire(s), gas layers and compartment surfaces (ceiling, walls and floor). This transfer is a function of the temperature differences and the emissivity of the gas layers as well as the compartment surfaces. For the fire and typical surfaces, emissivity values only vary over a small range. For the gas layers, however, the emissivity is a function of the concentration of species which are strong radiators: predominately smoke particulates, carbon dioxide, and water. Thus errors in the species concentrations can give rise to errors in the distribution of enthalpy among the layers, which results in errors in temperatures, resulting in errors in the flows. This illustrates just how tightly coupled the predictions made by CFAST can be.

1.6 Species Concentration and Deposition

When the layers are initialized at the start of the simulation, they are set to ambient conditions. These are the initial temperatures specified by the user, and 23 percent by mass (20.8 percent by volume) oxygen, 77 percent by mass (79 percent by volume) nitrogen, a mass concentration of water specified by the user as a relative humidity, and a zero concentration of all other species. As fuel is pyrolyzed, the various species are produced in direct relation to the mass of fuel burned (this relation is the species yield specified by the user for the fuel burning). Since oxygen is consumed rather than produced by the burning, the "yield" of oxygen is negative, and is set internally to correspond to the amount of oxygen needed to burn the fuel. Also, hydrogen cyanide and hydrogen chloride are assumed to be products of pyrolysis whereas carbon dioxide, carbon monoxide, water, and soot are products of combustion.

Each unit mass of a species produced is carried in the flow to the various rooms and accumulates in the layers. The model keeps track of the mass of each species in each layer, and knows the volume of each layer as a function of time. The mass divided by the volume is the mass concentration, which along with the molecular weight gives the concentration in volume percent or ppm as appropriate.

CFAST uses a combustion chemistry scheme different from any other model. While others compute each species concentration with an independent yield fraction, CFAST maintains a carbon-hydrogen-oxygen balance. The scheme is applied in three places. The first is burning in the portion of the plume which is in the lower layer of the room of fire origin. The second is the portion in the upper layer, also in the room of origin. The third is in the vent flow which entrains air from a lower layer into an upper layer in an adjacent compartment. This is equivalent to solving the conservation equations for each species independently.

1.7 Assumptions and Limitations

CFAST consists of a collection of data and computer programs which are used to *simulate* the important time-dependent phenomena involved in fires. The major functions provided include calculation of:

- the production of enthalpy and mass (smoke and gases) by one or more burning objects in one room, based on small- or large-scale measurements,

- the buoyancy-driven as well as forced transport of this energy and mass through a series of specified rooms and connections (e.g., doors, windows, cracks, ducts),

- the resulting temperatures, smoke optical densities, and gas concentrations after accounting for heat transfer to surfaces and dilution by mixing with clean air.

As can be seen from this list, fire modeling involves an interdisciplinary consideration of physics, chemistry, fluid mechanics, and heat transfer. In some areas, fundamental laws (conservation of mass, energy, and momentum) can be used, whereas in others empirical correlations or even "educated guesses" must be employed to bridge gaps in existing knowledge. The necessary approximations required by operational practicality result in the introduction of uncertainties in the results. The user should understand the inherent assumptions and limitations of the programs, and use these programs judiciously – including sensitivity analyses for the ranges of values for key parameters – in order to make estimates of these uncertainties. This section provides an overview of these assumptions and limitations.

1.7.1 Specified Fire Limitations

An important limitation of CFAST is the absence of a fire growth model. At the present time, it is not practical to adapt currently available fire growth models for direct inclusion in CFAST. Therefore, the system utilizes a user specified fire, expressed in terms of time specified rates of energy and mass released by the burning item(s). Such data can be obtained by measurements taken in large- and small-scale calorimeters, or from room burns. Examples of their associated limitations are as follows:

1. For a large-scale calorimeter, a product (e.g., chair, table, bookcase) is placed under a large collection hood and ignited by a 50 kW gas burner (simulating a wastebasket) placed adjacent to the item for 120 s. The combustion process then proceeds under assumed "free-burning" conditions, and the release rate data are measured. Potential sources of uncertainty here include measurement errors related to the instrumentation, and the degree to which "free-burning" conditions are not achieved (e.g., radiation from the gases under the hood or from the hood itself, and restrictions in the air entrained by the object causing locally reduced oxygen concentrations affecting the combustion chemistry). There are limited experimental data for upholstered furniture which suggest that prior to the onset of flashover in a compartment, the influence of the compartment on the burning behavior of the item is small. The differences obtained from the use of different types or locations of ignition sources have not been explored. These factors are discussed in reference [15].

2. Where small-scale calorimeter data are used, procedures are available to extrapolate to the behavior of a full-size item. These procedures are based on empirical correlations of data which exhibit significant scatter, thus limiting their accuracy. For example, for upholstered furniture, the peak heat release rates estimated by the "triangular approximation" method averaged 91 percent (range 46 to 103 percent) of values measured for a group of 26 chairs with noncombustible frames, but only 63 perecnt (range 46 to 83 percent) of values measured for a group of 11 chairs with combustible frames [16]. Also, the triangle neglects the "tails" of the curve; these are the initial time from ignition to significant burning of the item, and the region of burning of the combustible frame, after the fabric and filler are consumed.

3. The provided data and procedures only relate directly to burning of items initiated by relatively large flaming sources. Little data are currently available for release rates under smoldering combustion, or for the high external flux and low oxygen conditions characteristic of post-flashover burning. While the model allows multiple items burning simultaneously, it does not account for the synergy of such multiple fires. Thus, for other ignition scenarios, multiple items burning simultaneously (which exchange energy by radiation and convection), combustible interior finish, and post-flashover conditions, the move can give estimates which are often nonconservative (the actual release rates would be *greater* than estimated). At present, the only sure way to account for all of these complex phenomena is to conduct a full-scale room burn and use the pyrolysis rates directly. Subsequent versions of the model will include detailed combustion models which can be used as the source fire.

1.7.2 Zone Model and Transport Limitations

The basic assumption of all zone fire models is that each room can be divided into a small number of control volumes, each of which is internally uniform in temperature and composition. In CFAST, all rooms have two zones except the fire room, which has an additional zone for the fire plume. The boundary between the two layers in a room is called the interface.

It has generally been observed that in the spaces close to the fire, buoyantly stratified layers form. While in an experiment the temperature can be seen to vary within a given layer, these variations are small compared to the temperature difference between the layers.

Beyond the basic zone assumptions, the model typically involves a mixture of established theory (e.g., conservation equations), empirical correlations where there are data but no theory (e.g., flow and entrainment coefficients), and approximations where there are neither (e.g., post-flashover combustion chemistry) or where their effect is considered secondary compared to the "cost" of inclusion. An example of a widely used assumption is that the estimated error from ignoring the variation of the thermal properties of structural materials with temperature is small. While this information would be fairly simple to add to the computer code, data are scarce over a broad range of temperatures even for the most common materials.

With a highly complex model such as CFAST, the only reasonable method of assessing impacts of assumptions and limitations is through the verification process, which is ongoing at the Building and Fire Research Laboratory (BFRL). Until the results of this process are available, the user should be aware of the general limits of zone modeling and some specific manifestations in CFAST. These include the following:

1. Burning can be constrained by the available oxygen. However, this "constrained fire" (a "type 2" fire, see page 17) is not subject to the influences of radiation to enhance its burning rate, but is influenced by the oxygen available in the room. If a large mass loss rate is entered, the model will follow this input until there is insufficient oxygen available for that quantity of fuel to burn in the room. The unburned fuel (sometimes called excess pyrolyzate) is tracked as it flows out in the door jet, where it can entrain more oxygen. If this mixture is within the user-specified flammable range, it burns in the door plume. If not, it will be tracked throughout the building until it eventually collects as unburned fuel or burns in a vent. The enthalpy released in the fire room and in each vent, as well as the total enthalpy released, is detailed in the output of the model. Since mass and enthalpy are conserved, the total will be correct. However, since combustion did not take place adjacent to the burning object, the actual mass burned could be lower than that specified by the user. The difference will be the unburned fuel.

2. An oxygen combustion chemistry scheme is employed only in constrained (type 2) fires. Here user-specified hydrocarbon ratios and species yields are used by the model to predict concentrations. A balance among hydrogen, carbon, and oxygen molecules is maintained. Under some conditions, low oxygen can change the combustion chemistry, with an attendant increase in the yields of products of incomplete combustion such as CO. Guidance is provided on how to adjust the CO/CO_2 ratio. However, not enough is known about these chemical processes to build this relationship into the model at the present time. Some data exist in reports of full-scale experiments (e.g., reference [17]) which can assist in making such determinations.

3. The entrainment coefficients are empirically determined values. Small errors in these values will have a small effect on the fire plume or the flow in the plume of gases exiting the door of that room. In a multi-compartment model such as CFAST, however, small errors in each door plume are multiplicative as the flow proceeds through many compartments, possibly resulting in a significant error in the furthest rooms. The data available from validation experiments [18] indicate that the values for entrainment coefficients currently used in most zone models produce good agreement for a three-compartment configuration. More data are needed for larger numbers of rooms to study this further.

4. In real fires, smoke and gases are introduced into the lower layer of each room primarily due to mixing at connections between rooms and from the downward flows along walls (where contact with the wall cools the gas and reduces its buoyancy). Doorway mixing has been included in CFAST, using an empirically derived mixing coefficient. However, for smoke flow along a wall, the associated theory is only now being developed and is not included in the model. This may produce an underestimate of the lower layer concentrations. The most important manifestation of this underestimate will be the temperature distribution between the upper and lower layers caused by radiation.

5. The only mechanisms provided in zone models to move enthalpy and mass into the upper layer of a room are two types of plumes: those formed by the burning item(s) in the fire room, and those formed by the jet of upper layer gases flowing through an opening. Thus, when the model calculates the flow of warm, lower layer gases through a low opening (e.g., the undercut of a “closed” door) by expansion of the smoke layer, they are assigned to the lower layer of the room into which they flowed where they remain until the upper layer in the source room drops to the level of the undercut and the door jet forms. Thus, for a time the receiving room can show a lower layer temperature which exceeds that in the upper layer (a physically impossible condition). A better understanding of the flow within compartments in the context of a zone fire model would allow us to remove this anomaly. However, no hazard will exist during this time as the temperatures are low, and no gas species produced by the fire are carried through the opening until the upper layer drops to the height of the undercut.

2 Predictive Equations Used by the Model

This section presents a derivation of the predictive equations for zone fire models and explains in detail the ones used in CFAST [9], [11]. All current zone fire models take the mathematical form of an initial value problem for a mixed system of differential and algebraic equations. These equations are derived from the conservation of mass, energy and momentum. Subsidiary equations are the ideal gas law, and definitions of density and internal energy (for example, see [19]). These conservation laws are invoked for each zone or control volume. For further information on the numerical implications of these choices please see reference [20].

The basic element of the model is a zone. The basic assumption of a zone model is that properties such as temperature can be approximated throughout the zone by some uniform function. The usual approximation is that temperature, density and so on are uniform within a zone. This is not a necessary approximation. For example, a temperature which increases monotonically from the bottom of the zone to the top uniformly would, perhaps, improve the precision somewhat. However, the assumption of uniform properties is reasonable and yields good agreement with experiment. In general, these zones are grouped within compartments. The usual grouping is two gas layers per compartment. Once again, more could be utilized with a concomitant increase in computing time, but little improvement in accuracy.

There are two reasonable conjectures which dramatically improve the ease of solving these equations. Momentum is ignored within a compartment. The momentum of the interface has no significance in the present context. However, at boundaries such as windows, doors and so on, the Euler equation is integrated explicitly to yield the Bernoulli equation. This is solved implicitly in the equations which are discussed below. This stratagem avoids the short time step imposed by acoustic waves (Courant condition), which couple the pressure equation and the momentum equation. The other approximation is that the pressure is approximately uniform within a compartment. The argument is that a change in pressure of a few tens of Pascals over the height of the compartment is negligible in comparison with atmospheric pressure. Once again, this is applied to the basic conservation equations. This is consistent with the point source view of finite element models. Volume is merely one of the dependent variables. However, the hydrostatic variation in pressure *is* taken into account in calculating pressure differences between compartments. This might be viewed as analogous to the Bousinesque approximation in plume dynamics.

Many formulations based upon these assumptions can be derived. Several of these are discussed later. One formulation can be converted into another using the definitions of density, internal energy and the ideal gas law. Though equivalent analytically, these formulations differ in their numerical properties. Also, until the development of FAST [9], all models of this type assumed that the pressure equilibrated instantaneously, and thus the dP/dt term could be set to zero. However, as has been shown [21], it is better to solve these equations in the differential rather than the algebraic form if the proper solver is used.

As discussed in references [20] and [22], the zone fire modeling differential equations (ODE's) are stiff. Physically, the equations are stiff because of the presence of multiple time scales. Pressures adjust to changing conditions much quicker than other quantities such as layer temperatures or interface heights. Special solvers are required in general to solve zone fire modeling ODE's because of this stifness. Runge-Kutta methods or predictor-corrector methods such as Adams-Bashforth require prohibitively small time steps in order to track the short-time scale phenomena (pressure in our case). Methods that calculate the Jacobian (or at least approximate it) have a much larger stability region for stiff problems and are thus more successful at their solution.

Each formulation can be expressed in terms of mass and enthalpy flow. These rates represent the exchange of mass and enthalpy between zones due to physical phenomena such as plumes, natural and forced ventilation, convective and radiative heat transfer, and so on. For example, a vent exchanges mass and enthalpy between zones in connected rooms, a fire plume typically adds heat to the upper layer and transfers entrained mass and enthalpy from the lower to the upper layer, and convection transfers enthalpy from the gas layers to the surrounding walls.

We use the formalism that the mass flow to the upper and lower layers is denoted $\dot{m}_U$ and $\dot{m}_L$ and the enthalpy flow to the upper and lower layers is denoted $\dot{s}_U$ and $\dot{s}_L$. It is tacitly assumed that these rates may be computed in terms of zone properties such as temperature and density. These rates represent the net sum of all possible sources of mass and enthalpy due to phenomena such as those listed above. The numerical characteristics of the various formulations are easier to identify if the underlying physical phenomena are decoupled in this way.

Many approximations are necessary when developing physical sub-models for the mass and enthalpy terms. For example, most fire models assume that 1) the specific heat terms c_p and c_v are constant even though they depend upon temperature, 2) hydrostatic terms can be ignored in the equation of state (the ideal gas law) relating density of a layer with its temperature. However, the derivations which follow are all based on the basic conservation laws.

2.1 Derivation of Equations for a Two-Layer Model

A compartment is divided into two control volumes, a relatively hot upper layer and a relatively cooler lower layer, as illustrated in Figure 2. The gas in each layer has attributes of mass, internal energy, density, temperature, and volume denoted respectively by m_i, E_i, ρ_i, T_i, and V_i where $i=L$ for the lower layer and $i=U$ for the upper layer. The compartment as a whole has the attribute of pressure P. These 11 variables are related by means of the following seven constraints

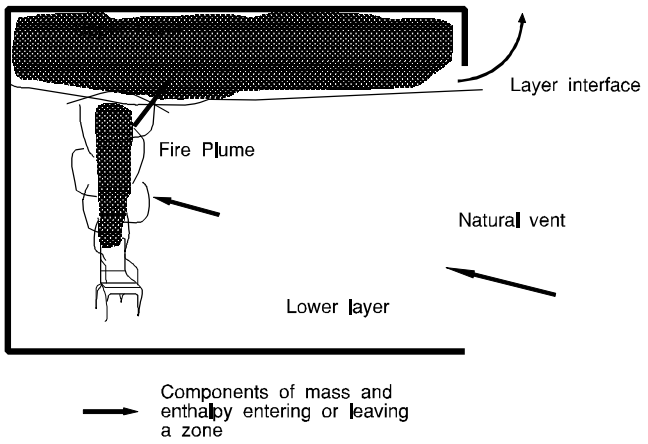


Figure 2. Schematic of control volumes in a two-layer zone model.

$$\rho_i = \frac{m_i}{V_i} \quad \text{(density)} \tag{1}$$

$$E_i = c_v m_i T_i \quad \text{(internal energy)} \tag{2}$$

$$P = R\rho_i T_i \quad \text{(ideal gas law)} \tag{3}$$

$$V = V_L + V_U \quad \text{(total volume)} \tag{4}$$

We get seven by counting density, internal energy and the ideal gas law twice (once for each layer). The specific heat at constant volume and at constant pressure c_v and c_p, the universal gas constant, R, and the ratio of specific heats, γ, are related by $\gamma = c_p / c_v$ and $R = c_p - c_v$. For air, $c_p \approx 1000$ kJ/kg K and $\gamma = 1.4$. This leaves four unconstrained, or independent, variables. So we require four equations for a unique solution. The four are the conservation of mass and enthalpy for each layer.

The differential equations for mass in each layer are trivially

$$\begin{aligned} \frac{dm_L}{dt} &= \dot{m}_L \\ \frac{dm_U}{dt} &= \dot{m}_U \end{aligned} \qquad (5)$$

The first law of thermodynamics states that the rate of increase of internal energy plus the rate at which the layer does work by expansion is equal to the rate at which enthalpy* is added to the gas. In differential form this is

$$\begin{array}{ccccc} \text{internal energy} & + & \text{work} & = & \text{enthalpy} \\ \overbrace{\phantom{\frac{dE_i}{dt}}} & & \overbrace{\phantom{P\frac{dV_i}{dt}}} & & \overbrace{\phantom{\dot{h}_i}} \\ \frac{dE_i}{dt} & + & P\frac{dV_i}{dt} & = & \dot{h}_i \end{array} \qquad (6)$$

A differential equation for pressure can be derived by adding the upper and lower layer versions of eq (6), noting that $dV_U/dt = -dV_L/dt$, and substituting the differential form of eq (2) to yield

$$\frac{dP}{dt} = \frac{\gamma - 1}{V}\left(\dot{h}_L + \dot{h}_U\right). \qquad (7)$$

Differential equations for the layer volumes can be obtained by substituting the differential form of eq (2) into eq (6) to obtain

$$\frac{dV_i}{dt} = \frac{1}{P\gamma}\left((\gamma - 1)\,\dot{h}_i - V_i\,\frac{dP}{dt}\right). \qquad (8)$$

Equation (6) can be rewritten using eq (8) to eliminate dV/dt to obtain

$$\frac{dE_i}{dt} = \frac{1}{\gamma}\left(\dot{h}_i + V_i\frac{dP}{dt}\right). \qquad (9)$$

---

* The term enthalpy is generalized from a rigorous definition. For vent flow, the term enthalpy is correctly used to indicate the energy flow and work term. For other phenomena, only the energy term is included; the work term is presumed to be zero in the context of a zone model.

A differential equation for density can be derived by applying the quotient rule to $\frac{d\rho_i}{dt} = \frac{d}{dt}\left(\frac{m_i}{V_i}\right)$ and using eq (8) to eliminate dV_i/dt to obtain

$$\frac{d\rho_i}{dt} = -\frac{1}{c_p T_i V_i}\left(\left(\dot{h}_i - c_p \dot{m}_i T_i\right) - \frac{V_i}{\gamma - 1}\frac{dP}{dt}\right) . \tag{10}$$

Temperature differential equations can be obtained from the equation of state by applying the quotient rule to $\frac{dT_i}{dt} = \frac{d}{dt}\left(\frac{P}{R\rho_i}\right)$ and using eq (10) to eliminate $d\rho/dt$ to obtain

$$\frac{dT_i}{dt} = \frac{1}{c_p \rho_i V_i}\left(\left(\dot{h}_i - c_p \dot{m}_i T_i\right) + V_i \frac{dP}{dt}\right) . \tag{11}$$

These equations for each of the eleven variables are summarized in Table 1. The time evolution of these solution variables can be computed by solving the corresponding differential equations together with appropriate initial conditions. The remaining seven variables can be determined from the four independent solution variables using eqs (1) to (4).

There are, however, many possible differential equation formulations. Indeed, there are 330 different ways to select four variables from eleven. Many of these systems are incomplete due to the relationships that exist between the variables given in eqs (1) to (4). For example the variables, ρ_U, V_U, m_U, and P form a dependent set since $\rho_U = m_U / V_U$. Table 2 shows the solution variable selection made by several zone fire models.

The number of differential equation formulations can be considerably reduced by not mixing variable types between layers; that is, if upper layer mass is chosen as a solution variable, then lower layer mass must also be chosen. For example, for two of the solution variables choose m_L and m_U, or ρ_L and ρ_U, or T_L and T_U. For the other two solution variables pick E_L and E_U or P and V_L or P and V_U. This reduces the number of distinct formulations to nine. Since the numerical properties of the upper layer volume equation are the same as a lower layer one, the number of distinct formulations can be reduced to six.

Table 1. Conservative Zone Modeling Differential Equations

Equation Type	Differential Equation
i'th layer mass	$\frac{dm_i}{dt} = \dot{m}_i$
pressure	$\frac{dP}{dt} = \frac{\gamma - 1}{V}\left(\dot{h}_L + \dot{h}_U\right)$
i'th layer energy	$\frac{dE_i}{dt} = \frac{1}{\gamma}\left(\dot{h}_i + V_i\frac{dP}{dt}\right)$
i'th layer volume	$\frac{dV_i}{dt} = \frac{1}{\gamma P}\left((\gamma - 1)\dot{h}_i - V_i\frac{dP}{dt}\right)$
i'th layer density	$\frac{d\rho_i}{dt} = -\frac{1}{c_p T_i V_i}\left(\left(\dot{h}_i - c_p\dot{m}_i T_i\right) - \frac{V_i}{\gamma - 1}\frac{dP}{dt}\right)$
i'th layer temperature	$\frac{dT_i}{dt} = \frac{1}{c_p\rho_i V_i}\left(\left(\dot{h}_i - c_p\dot{m}_i T_i\right) + V_i\frac{dP}{dt}\right)$

Table 2. Conservative Zone Model Equation Selections

Zone Fire Model	Equations	Substitutions
FAST	$\frac{d\Delta P}{dt}$, $\frac{dV_L}{dt}$, $\frac{dT_U}{dt}$, $\frac{dT_L}{dt}$	$\Delta P = P - P_{ref}$
CCFM.HOLE	$\frac{d\Delta P}{dt}$, $\frac{dy}{dt}$, $\frac{d\rho_U}{dt}$, $\frac{d\rho_L}{dt}$	$\Delta P = P - P_{ref}$ $y = V_L / A_{room}$
CCFM.VENTS	$\frac{d\Delta P}{dt}$, $\frac{dy}{dt}$, $\frac{dm_U}{dt}$, $\frac{dm_L}{dt}$	$\Delta P = P - P_{ref}$ $y = V_L / A_{room}$
FIRST, HARVARD	$\frac{dE_U}{dt}$, $\frac{dE_L}{dt}$, $\frac{dm_U}{dt}$, $\frac{dm_L}{dt}$	

2.2 Equation Set Used in CFAST

The current version of CFAST is set up to use the equation set for layer temperature, layer volume, and pressure as shown below. However, the internal structure of the model is such that it will allow any of the formulations above to be substituted with minimal effort.

$$P = P_{ref} + \Delta P \tag{12}$$

$$\frac{dP}{dt} = \frac{\gamma - 1}{V}\left(\dot{h}_L + \dot{h}_U\right) \tag{13}$$

$$\frac{dV_U}{dt} = \frac{1}{\gamma P}\left((\gamma - 1)\dot{h}_U - V_U\frac{dP}{dt}\right) \tag{14}$$

$$\frac{dT_U}{dt} = \frac{1}{c_p\rho_U V_U}\left(\left(\dot{h}_U - c_p\dot{m}_U T_U\right) + V_U\frac{dP}{dt}\right) \tag{15}$$

$$\frac{dT_L}{dt} = \frac{1}{c_p\rho_L V_L}\left(\left(\dot{h}_L - c_p\dot{m}_L T_L\right) + V_L\frac{dP}{dt}\right) \tag{16}$$

3 Source Terms

The conserved quantities in each compartment are described by the set of predictive equations shown above. The form of the equations is such that the physical phenomena are source terms on the right-hand-side of these equations. Such a formulation makes the addition and deletion of physical phenomena and changing the form of algorithms a *relatively* simple matter. For each of the phenomena discussed below, the physical basis for the model is discussed first, followed by a brief presentation of the implementation within CFAST. For all of the phenomena, there are basically two parts to the implementation: the physical interface routine (which is the interface between the CFAST model and the algorithm) and the actual physical routine(s) which implement the physics. This implementation allows the physics to remain independent of the structure of CFAST and allows easier insertion of new phenomena.

3.1 The Fire

3.1.1 Specified Fire (Fire Types 1 and 2)

A specified fire is one for which the time dependent characteristics are specified as a function of time. The specified fire can be unconstrained or constrained. These fires are later referred to as type 1 and type 2, respectively. The meaning of this assignment will become clearer in the discussion of the data file structure. For the constrained fire, the constraint is based on the minimum of the fuel and oxygen available for combustion. For either, the pyrolysis rate is specified as $\dot{m}_f$, the burning rate as $\dot{m}_b$ and the heat of combustion as H_c so that the nominal heat release rate is

$$\dot{Q}_f = H_c \dot{m}_b - c_p \left(T_u - T_v\right) \dot{m}_b \ . \tag{17}$$

For the unconstrained fire, $\dot{m}_b = \dot{m}_f$, whereas for the constrained fire, the burning rate may be less than the pyrolysis rate. Models of specified fires generally use an effective heat of combustion which is obtained from an experimental apparatus such as the Cone Calorimeter [23]. The shortcoming of this approach is that the pyrolysis rate is not connected to radiative feedback from the flame or compartment. In an actual fire, this is an important consideration, and the specification used should match the experimental conditions as closely as possible.

The enthalpy which is released goes into radiation and convection

$$\begin{aligned} \dot{Q}_r(fire) &= \chi_R \dot{Q}_f \\ \dot{Q}_c(fire) &= \left(1 - \chi_r\right) \dot{Q}_f \ . \end{aligned} \tag{18}$$

The term $Q_c(fire)$ then becomes the driving term in the plume flow. In the actual implementation, these formulae are modified to be compatible with the two-zone nature of the model. For a specified fire there is radiation to both the upper and lower layers, whereas the convective part contributes only to the upper layer. In addition, a view factor must be calculated for the radiative portion.

3.1.2 Combustion Chemistry (Fire Type 2)

For a type 2 fire, the products of combustion are calculated via a species balance consistent with the constraint on available oxygen. The scheme is applied in three places. The first is burning in the portion of the plume which is in the lower layer of the room of fire origin (region #1). The second is the portion in the upper layer, also in the room of origin (region #2). The third is in the vent flow which entrains air from a lower layer into an upper layer in an adjacent compartment (region #3). Figure 3 is a schematic of the concept of division of burning regions.

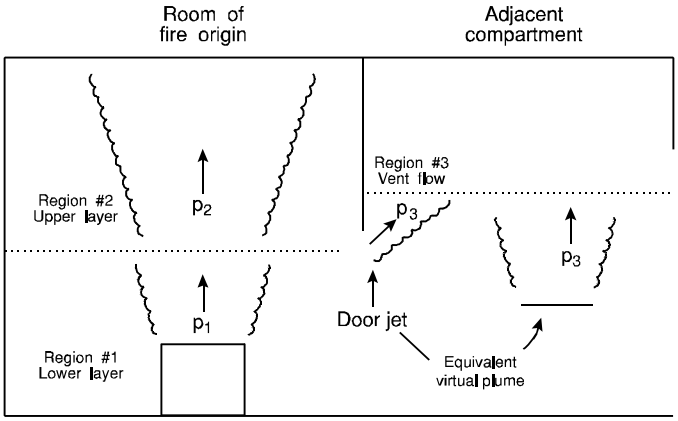


Figure 3. Schematic of entrainment and burning regions.

The simplest form of enthalpy release is made by specifying a heat release rate, together with a consistent pyrolysis rate. This would simulate the fire that occurs in an unconfined area. This is the form that all zone models have used until now. As soon as one is constrained by the confines of a compartment, then the nature of the fire changes. In particular, the available oxygen may not be sufficient to allow complete combustion. It is not sufficient to account for the oxygen alone, however. A consistent prescription for the combustion chemistry is required. In CFAST, we calculate a chemical balance for a simplified combustion reaction in terms of the ratios of species to carbon dioxide, the primary product of combustion. We allow for a realistic fuel composition, including oxygen, carbon, hydrogen, and chlorine as part of the fuel. Carbon monoxide, carbon dioxide, soot, water, hydrogen cyanide and hydrogen chloride are the products of combustion. Infinite rate kinetics is assumed. Further, we do not calculate backwards reactions.

The fuel burning rate in terms of the carbon production is

$$\dot{m}_f = \{-\} \times \dot{m}_c \tag{19}$$

where {—} is the multiplier in the fuel production

$$\{-\} = \left(1 + \frac{H}{C} + \frac{HCl}{C} + \frac{HCN}{C} + \frac{O}{C}\right) \equiv f/C \ . \tag{20}$$

The following definitions are for the heat release rate as a function of the fuel burning rate, and the heat release rate based on oxygen consumption. *H/C, HCl/C, HCN/C* and *O/C* are the ratio of mass of that species to carbon in the fuel. Thus H/C is for the mass of hydrogen to the mass of carbon produced in **pyrolysis**. This is a very useful way to characterize the fuel. This is in terms of the elemental composition of the fuel, and not elemental molecules, such as H_2. These are the ratios for the fuel, and the material which comes from it. For the products of the combustion process, we have CO_2/C, CO/C, H_2O/C and S/C. These ratios are in terms of free molecules, generally gaseous. Note that the "S" is used to designate soot, and we assume it consists primarily of carbon.

The first step is to limit the actual burning which takes place in the combustion zone. In each combustion zone, there is a quantity of fuel available. At the source this results from the pyrolysis of the material. In other situations such as a plume or door jet, it is the net unburned fuel available. At the source we refer to $\dot{m}_p$,

but in the other regions we use a catch-all, $\dot{m}_{TUHC}$ (for total unburned hydrocarbons). In each case, the fuel which is available but not burned is then deposited into this category. This provides a consistent notation. In the discussion below, we will speak of the $\dot{m}_f$ as the amount burned. The understanding is that in the iterative scheme discussed, this value is initialized to the available fuel, and then possibly reduced by the algorithm discussed. Subsequently, the available fuel, $\dot{m}_{TUHC}$, is reduced by the final value of $\dot{m}_f$. Thus we have a consistent description in each burning region, with an algorithm that can be invoked independent of the region being analyzed. The energy released by the available fuel if there were no constraint (free burn) is

$$\dot{Q} = \dot{m}_f \times H_c \ , \tag{21}$$

with the mass of oxygen required to achieve this energy release rate (based on the oxygen consumption principle [24]) of

$$\dot{m}_O = \frac{\dot{Q}}{1.32\times10^7} = \dot{m}_f \times \frac{H_c}{1.32\times10^7} \ . \tag{22}$$

If the fuel contains oxygen, the oxygen needed to achieve full combustion is less than this value

$$\dot{m}_O(\text{needed}) = \dot{m}_O - \dot{m}_O(\text{in the fuel}) \tag{23}$$

If sufficient oxygen is available, then it is fully burned. However, if the oxygen concentraion is low enough, it will constrain the burning and impose a limit on the the amount of fuel actually burned, as opposed to the amount pyrolyzed. The actual limitation is discussed below and is presented as eq (38).

$$\dot{m}_O(\text{actual}) = \text{minimum of} \left\{ \dot{m}_O(\text{available}),\ \dot{m}_O(\text{needed}) \right\}, \tag{24}$$

$$\dot{m}_f(\text{actual}) = \dot{m}_O(\text{actual}) \times \frac{1.32\,x10^7}{H_c} \tag{25}$$

Essentially, we limit the amount of fuel that is burned, as opposed to the amount that is pyrolyzed, to the lesser of the amount pyrolyzed and that required to consume the *available* oxygen. The $\dot{m}_O$(actual) and $\dot{m}_f$(actual) are the quantities used below.

We begin with the mass balance equation. The mass consumed as pyrolyzate plus oxygen must reappear as product.

$$\begin{aligned} \dot{m}_f + \dot{m}_O &= \dot{m}_f + \dot{m}_f \times \frac{H_c}{1.32\times10^7} - \frac{\dot{m}_f}{\{-\}} \times \left(\frac{O}{C} \right) \\ &= \dot{m}_{CO_2} + \dot{m}_{CO} + \dot{m}_S + \dot{m}_{H_2O} + \dot{m}_{HCl} + \dot{m}_{HCN} \end{aligned} \tag{26}$$

We then substitute the following definitions of mass produced of each species based on the amount of carbon consumed as

$$\dot{m}_{HCl} = \left(\frac{HCl}{C}\right) \times \dot{m}_C \rightarrow \left(\frac{HCl}{f}\right) \times \dot{m}_f \tag{27}$$

$$\dot{m}_{HCN} = \left(\frac{HCN}{C}\right) \times \dot{m}_C \rightarrow \left(\frac{HCN}{f}\right) \times \dot{m}_f \tag{28}$$

$$_{2}O = \frac{1}{2}\left(\frac{H_2O}{H}\right) \times \left(\frac{H}{C}\right) \times \dot{m}_C = 9 \times \left(\frac{H}{C}\right) \times \dot{m}_C \rightarrow 9 \times \left(\frac{H}{C}\right) \times \frac{\dot{m}}{\{-}$$ (29)

$$\dot{m}_{CO_2} = \left(\frac{CO_2}{C}\right) \times \dot{m}_C \tag{30}$$

$$\dot{m}_S = \left(\frac{S}{C}\right) \times \dot{m}_C = \left(\frac{CO_2}{C}\right) \times \left(\frac{S}{CO_2}\right) \times \dot{m}_C \rightarrow \left(\frac{S}{CO_2}\right) \times \dot{m}_{CO_2} \tag{31}$$

$$\dot{m}_{CO} = \left(\frac{CO}{C}\right) \times \dot{m}_C = \left(\frac{CO_2}{C}\right) \times \left(\frac{CO}{CO_2}\right) \times \dot{m}_C \rightarrow \left(\frac{CO}{CO_2}\right) \times \dot{m}_{CC}$$ (32)

Substituting the above definitions into the mass balance equation yields:

$$\left(\frac{CO_2}{C}\right) = \frac{\{-\} \times \left(1 + \frac{H_c}{1.32\times10^7} - \frac{O/C}{\{-\}}\right) - \left(\frac{HCl}{C} + \frac{HCN}{C} + \frac{H}{C}\right.}{\left(1 + \frac{S}{CO_2} + \frac{CO}{CO_2}\right)} \tag{33}$$

With this definition, we can substitute back into the equation for carbon dioxide production, which yields

$$_{2} = \dot{m}_f \times \frac{\left(1 + \frac{h_c}{1.32\times10^7} - \frac{O/C}{\{-\}}\right) - \left(\frac{HCl}{C} + \frac{HCN}{C} + \frac{H}{C}\right)/\{-}}{\left(1 + \frac{S}{CO_2} + \frac{CO}{CO_2}\right)}$$ (34)

The form in which we cast these equations evolves naturally from the properties of combustion. Hydrogen, carbon and bound oxygen are properties of the fuel. They can be measured experimentally independent of the combustion process. Thus we use these ratios as the basis of the scheme. In a similar sense, hydrogen chloride and hydrogen cyanide are properties of the pyrolysis process. So hydrogen chlorine and hydrogen cyanide production are specified with respect to the fuel pyrolysis. Normally this is how they are measured, for example with the cone calorimeter, so we can use the measured quantities directly. Other than the cyanide, chloride and water production, hydrogen does not play a role. In general, hydrogen has much more

of an affinity for oxygen than for carbon, so almost all of the hydrogen will go toward production of water. This dictates the next choice, which is that soot is essentially all carbon. On a mass basis this is certainly true. On a molecular basis, however, it may consist of molecules which vary greatly in size. Carbon dioxide is a direct product of combustion, and the assumption is that most carbon will end up here. Thus, in this model, carbon monoxide and soot are functions of incomplete combustion. They are assumed not to be a function of the pyrolysis process itself. (Although carbon monoxide can be produced directly, it is presumed such production will be negligible when compared to its generation during incomplete combustion.) Thus they depend on the environment in which the burning takes place. Thus the production of these products are specified with respect to the carbon dioxide. At present, we must rely on measured ratios, but this is beginning to change as we gain a better understanding of the combustion process. So, in the present model, carbon goes to one of three final species, carbon dioxide, carbon monoxide or soot (carbon), with the particular branching ratio depending on the chemistry active at the time.

Equations (29) through (34) are used in terms of the carbon production. We now need to recast HCl and HCN in terms of fuel production rather than carbon production, since that is how they are measured. Since HCl and HCN are similar, we will just make the argument for one, and then assume that the derivation is the same. One simplification will be possible for the HCN though, and that is that its production rate is *always* much less than the pyrolysis rate.

Since {—} is just f/C,

$$\left\{\frac{HCl}{C}\right\} = \left\{\frac{HCl}{f}\right\} \times \left(1 + \frac{H}{C} + \frac{HCl}{C} + \frac{HCN}{C} + \frac{O}{C}\right). \tag{35}$$

Therefore

$$\left\{\frac{HCl}{C}\right\} = \left\{\frac{HCl}{f}\right\} \times \left(\frac{1 + \frac{H}{C} + \frac{O}{C}}{1 - \left\{\frac{HCl}{f}\right\}}\right), \tag{36}$$

and for hydrogen cyanide we have

$$\left\{\frac{HCN}{C}\right\} = \left\{\frac{HCN}{f}\right\} \times \left(1 + \frac{H}{C} + \frac{HCl}{C} + \frac{O}{C}\right). \tag{37}$$

In this latter case, we assume that the cyanide ratio (*HCN/C*) is small compared to unity. It is the *HCl/C* and *HCN/C* ratios which are used by the model.

The relationship between oxygen and fuel concentration defines a range where burning will take place. The rich limit is where, for a given ratio of O_2 to N_2 (generally the ratio in air), there is too much fuel for combustion. At the other end, there is the lean flammability limit, where there is too little fuel for combustion. In the CFAST model, the rich limit is incorporated by limiting the burning rate as the oxygen level decreases until a "lower oxygen limit" is reached. The lower oxygen limit is incorporated through a smooth decrease in the burning rate near the limit:

$$\dot{m}_o(available) = \dot{m}_e Y_{O_2} C_{LOL} \tag{38}$$

The lower oxygen limit coefficient, C_{LOL}, is the fraction of the available fuel which can be burned with the available oxygen and varies from 0 at the limit to 1 above the limit. The functional form provides a smooth cutoff of the burning over a narrow range above the limit.

$$C_{LOL} = \frac{\tanh\left(800\left(Y_{O_2} - Y_{LOL}\right) - 4\right) + 1}{2} \tag{39}$$

For the lean flammability limit, an ignition temperature criterion is included, below which no burning takes place.

In summary, the formation of some of the products of combustion, carbon dioxide, carbon monoxide, soot, water, hydrogen cyanide, and hydrogen chloride can be predicted given the branching ratios CO/CO_2, $S(soot)/CO_2$, the composition of the fuel, H/C, O/C, HCl/f and HCN/f and the flammability limit. In principle, the flammability limit comes from theory. At present, in practice we use experimental values, such as those from Morehart et al. [25]. The composition of the fuel is a measurable quantity, although it is complicated somewhat by physical effects. The complication arises in that materials such as wood will yield methane in the early stages of burning, and carbon rich products at later times. Thus the H/C and O/C ratios are functions of time. Finally, the production ratios of CO/CO_2, $S(soot)/CO_2$ are based on the kinetics which in turn is a function of the ambient environment.

Implementation (FIRES, DOFIRE, CHEMIE): The physical interface routine, FIRES, calculates the rates of addition of mass, enthalpy, and species into all layers in all rooms from all fires in a simulation. For each fire, the following scheme is employed:

1. For each specified fire, the routine PYROLS (for the main fire) or OBJINT (for other fires) calculates time dependent quantities for the time of interest by interpolating between the time points specified by the user. The routine FIRPLM calculates the plume entrainment rate.

2. For a type 1 (unconstrained) fire, the routine DOFIRE sets the burning rate to the pyrolysis rate. The heat release rate is found by multiplying the burning rate by the heat of combustion.

3. For a type 2 (constrained) fire, the prescribed chemistry scheme discussed above is used to constrain the burning rate based on *both* the fuel and oxygen available. This chemistry scheme is implemented in the routine CHEMIE. This calculation is done for both the lower layer (from the mass entrained by the plume) and for burning in the upper layer (with oxygen and fuel available in the layer).

3.2 Plumes

Buoyancy generated by the combustion processes in a fire causes the formation of a plume. Such a plume can transport mass and enthalpy from the fire into the lower or upper layer of a compartment. In the present implementation, we assume that both mass and enthalpy from the fire are deposited only into the upper layer. In addition the plume entrains mass from the lower layer and transports it into the upper layer. This yields a net enthalpy flux between the two layers. Actually, the flame and plume will generally radiate somewhat into the lower layer, at least if it is not diathermous. So our approximation causes the upper layer to be somewhat hotter, and the lower layer somewhat cooler than is the case, at least in a well developed fire. For normal fires and door jet fires, plume entrainment is implemented as part of the fire calculation detailed in section 3.1.2.

A fire generates energy at a rate Q. Some fraction, χ_R, will exit the fire as radiation. The remainder, χ_C, will then be deposited in the layers as convective energy or heat additional fuel so that it pyrolyses. Defining this quantity ($C_p m_e(T_u-T_e)$) to be the convective heat release rate, we can use the work of McCaffrey [26] to estimate the mass flux from the fire into the upper layer. This correlation divides the flame/plume into three regions as shown below. This prescription agrees with the work of Cetegen et al. [27] in the intermittent regions but yields greater entrainment in the other two regions. This difference is particularly important for the initial fire since the upper layer is far removed from the fire.

$$\begin{aligned}
&\text{flaming:} && \frac{\dot{m}_e}{\dot{Q}} = 0.011 \left(\frac{Z}{\dot{Q}^{2/5}} \right)^{0.566} && 0.00 \le \left(\frac{Z}{\dot{Q}^{2/5}} \right) < 0.08 \\
&\text{intermittent:} && \frac{\dot{m}_e}{\dot{Q}} = 0.026 \left(\frac{Z}{\dot{Q}^{2/5}} \right)^{0.909} && 0.08 \le \left(\frac{Z}{\dot{Q}^{2/5}} \right) < 0.20 \\
&\text{plume:} && \frac{\dot{m}_e}{\dot{Q}} = 0.124 \left(\frac{Z}{\dot{Q}^{2/5}} \right)^{1.895} && 0.20 \le \left(\frac{Z}{\dot{Q}^{2/5}} \right)
\end{aligned} \qquad (40)$$

McCaffrey's correlation is an extension of the common point source plume model, with a different set of coefficients for each region. These coefficients are experimental correlations, and are not based on theory. The theory appears only in the form of the fitted function. The binding to the point source plume model is for the value for Z where the mode changes, namely from flaming to intermittent to plume.

Within CFAST, the radiative fraction defaults to 0.15; i.e., 15 percent of the fires energy is released via radiation. This value is consistent with typical methane flames. For other fuels, the work or Tewarson [28], McCaffrey [29], or Koseki [30] is available for reference. These place the typical range for the radiative fraction at a maximum of about 0.6.

In CFAST, there is a constraint on the quantity of gas which can be entrained by a plume arising from a fire. The constraint arises from the physical fact that a plume can rise only so high for a given size of a heat source. In the earlier versions of this model (FAST version 17 and earlier), the plume was not treated as a separate zone. Rather we assumed that the upper layer was connected immediately to the fire by the plume. The implication is that the plume is formed instantaneously and stretches from the fire to the upper layer or ceiling. Consequently, early in a fire, when the energy flux was very small and the plume length very long, the entrainment was over predicted. This resulted in the interface falling more rapidly than was seen in experiments. Also the initial temperature was too low and the rate of rise too fast, whereas the asymptotic temperature was correct. The latter occurred when these early effects were no longer important.

The correct sequence of events is for a small fire to generate a plume which does not reach the ceiling or upper layer initially. The plume entrains enough cool gas to decrease the buoyancy to the point where it no longer rises. When there is sufficient energy present in the plume, it will penetrate the upper layer. The effect is two-fold: first, the interface will take longer to fall and second, the rate of rise of the upper layer temperature will not be as great. To this end the following prescription has been incorporated: for a given size fire, a limit is placed on the amount of mass which can be entrained, such that no more is entrained than would allow the plume to reach the layer interface. The result is that the interface falls at about the correct rate, although it starts a little too soon, and the upper layer temperature is over predicted, but follows experimental data after the initial phase (see sec. 5).

For a section (segment) of the plume to penetrate the inversion formed by a hot layer at T_u over a cool layer T_l, the density of the gas in the plume at the point of intersection must be less than the density of the gas in

the upper layer, that is $\rho_p<\rho_u$. The subscript "v" is the virtual point at which mass is coming off the fire, "q" the state at which this same mass would be if were there no cooling from the entrained gases and "p" the plume at the point at which it intersects the upper layer. The "l" refers to the lower layer, and the "u" to the upper layer. The temperature rise in the plume is given by

$$(T_q - T_v)\, C_p \dot{m}_v = \dot{Q}. \tag{41}$$

From conservation of mass we have

$$\dot{m}_p = \dot{m}_e + \dot{m}_q. \tag{42}$$

And from conservation of enthalpy we have

$$\dot{m}_p T_p = \dot{m}_e T_l + \dot{m}_v T_v. \tag{43}$$

The criterion that the density in the plume region must be lower than the upper layer implies

$$T_u < T_p. \tag{44}$$

By substituting the equation for temperature rise, eq (41), and the conservation of mass, eq (42), into eq (43)

$$\dot{m}_e < \frac{\dot{Q}}{C_p(T_u - T_l)} - \left(\frac{T_u - T_v}{T_u - T_l} \right) \dot{m}_v. \tag{45}$$

The right most term is negligible in the cases under consideration so we will ignore it. It the case where it is of the same order as the first term, there are other constraints on the entrainment. Thus we are left with the maximum for $\dot{m}_e$ of

$$\dot{m}_e < \frac{\dot{Q}}{c_p(T_u - T_l)} \tag{46}$$

which is incorporated into the model. It should be noted that both the plume and layers are assumed to be well mixed with negligible mixing and transport time for the plume and layers.

3.3 Vent Flows

Mass flow (in the remainder of this section, the term "flow" will be used to mean mass flow) is the dominant source term for the predictive equations because it fluctuates most rapidly and transfers the greatest amount of enthalpy on an instantaneous basis of all the source terms. Also, it is most sensitive to changes in the environment. Flow through vents comes in two varieties. The first is horizontal flow. It is the flow which is normally thought of in discussing fires. It encompasses flow through doors, windows and so on. The other is vertical flow and can occur if there is a hole in the ceiling or floor of a compartment. This latter phenomena is particularly important in two disparate situations: a ship, and the role of fire fighters doing roof venting. Vertical flow is discussed in section 3.3.2.

3.3.1 Horizontal Flow Through Vertical Vents

Theory: Flow through normal vents is governed by the pressure difference across a vent. A momentum equation for the zone boundaries is not solved directly. Instead momentum transfer at the zone boundaries is included by using an integrated form of Euler's equation, namely Bernoulli's solution for the velocity equation. This solution is augmented for restricted openings by using flow coefficients [31] to allow for constriction from finite size doors. The flow (or orifice) coefficient is an empirical term which addresses the problem of constriction of velocity streamlines at an orifice.

Bernoulli's equation is the integral of the Euler equation and applies to general initial and final velocities and pressures. The implication of using this equation for a zone model is that the initial velocity in the doorway is the quantity sought, and the final velocity in the target compartment vanishes. That is, the flow velocity vanishes where the final pressure is measured. Thus, the pressure at a stagnation point is used. This is consonant with the concept of uniform zones which are completely mixed and have no internal flow. The general form for the velocity of the mass flow is given by

$$v = C\left(\frac{2\delta P}{\rho}\right)^{1/2} \tag{47}$$

where C is the constriction (or flow) coefficient (≈ 0.7), ρ is the gas density on the source side, and δP is the pressure across the interface. (Note: at present we use a constant C for all gas temperatures) We apply the above equation to rectangular openings which allows us to remove the width from the mass flux integral. That is

$$mass\ flux = \int_{width}\int_{height} \rho v dz dw \rightarrow width \int_{z_1}^{z_2} \rho v dz \ . \tag{48}$$

The simplest means to define the limits of integration is with neutral planes, that is the height at which flow reversal occurs, and physical boundaries such as sills and soffits. By breaking the integral into intervals defined by flow reversal, a soffit, a sill, or a zone interface, the flow equation can be integrated piecewise analytically and then summed.

The approach to calculating the flow field is of some interest. The flow calculations are performed as follows. The vent opening is partitioned into at most six slabs where each slab is bounded by a layer height, neutral plane, or vent boundary such as a soffit or sill. The most general case is illustrated in Figure 4.

The mass flow for each slab can be determined from

$$\dot{m}_{i\rightarrow o} = \frac{1}{3} C(8\rho) A_{slab}\left(\frac{|P_t| + xy + |P_b|}{x+y}\right) \tag{49}$$

where $x = |P_t|^{1/2}$, and $y = |P_b|^{1/2}$. P_t and P_b are the cross-vent pressure differential at the top and bottom of the slab respectively and A_{slab} is the cross-sectional area of the slab. The value of the density, ρ, is taken from the source compartment.

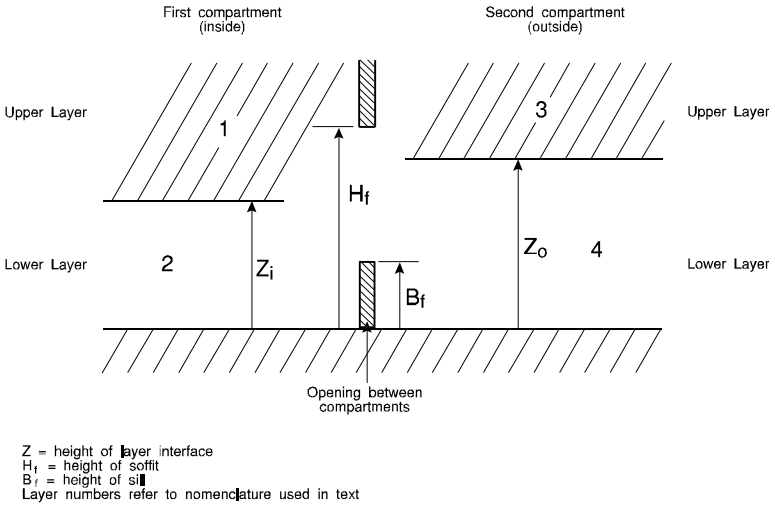


Figure 4. Notation conventions for two-layer model in two rooms with a connecting vent.

A mixing phenomenon occurs at vents which is similar to entrainment in plumes. As hot gases from one compartment leave that compartment and flow into an adjacent compartment a door jet can exist which is analogous to a normal plume. Mixing of this type occurs for $\dot{m}_{13} > 0$ as shown in Figure 5. To calculate the entrainment ($\dot{m}_{43}$ in this example), once again we use a plume description, but with an extended source point. The estimate for the point source extension is given by Cetegen et al. [27]. This virtual point source is chosen so that the flow at the door opening would correspond to a plume with the heating (with respect to the lower layer) given by

$$\dot{Q}_{eq} = c_p(T_1 - T_4)\dot{m}_{13} \tag{50}$$

The concept of the virtual source is that the enthalpy flux from the virtual point source should equal the actual enthalpy flux in the door jet at the point of exit from the vent using the same prescription. Thus the entrainment is calculated the same way as was done for a normal plume. The height, z_p, of the plume is

$$z_p = \frac{z_{13}}{\dot{Q}_{eq}^{2/5}} + v_p \tag{51}$$

where v_p, the virtual source point, is defined by inverting the entrainment process to yield

$$\begin{aligned}
v_p &= \left(\frac{90.9\dot{m}}{\dot{Q}_{eq}}\right)^{1.76} && \text{if } 0.00 < v_p \le 0.08 \\
v_p &= \left(\frac{38.5\dot{m}}{\dot{Q}_{eq}}\right)^{1.001} && \text{if } 0.08 < v_p \le 0.20 \\
v_p &= \left(\frac{8.10\dot{m}}{\dot{Q}_{eq}}\right)^{0.528} && \text{if } 0.20 < v_p
\end{aligned} \tag{52}$$

The units of this height, z_p and of v_p, are not length, but rather the reduced notation of McCaffrey [26]. That is, the z_p defined here is the term $z/Q^{2/5}$ used earlier. Although outside of the normal range of validity of the plume model, a level of agreement with experiment is apparent (see sec. 5). Since a door jet forms a flat plume whereas a normal fire plume will be approximately circular, strong agreement is not expected.

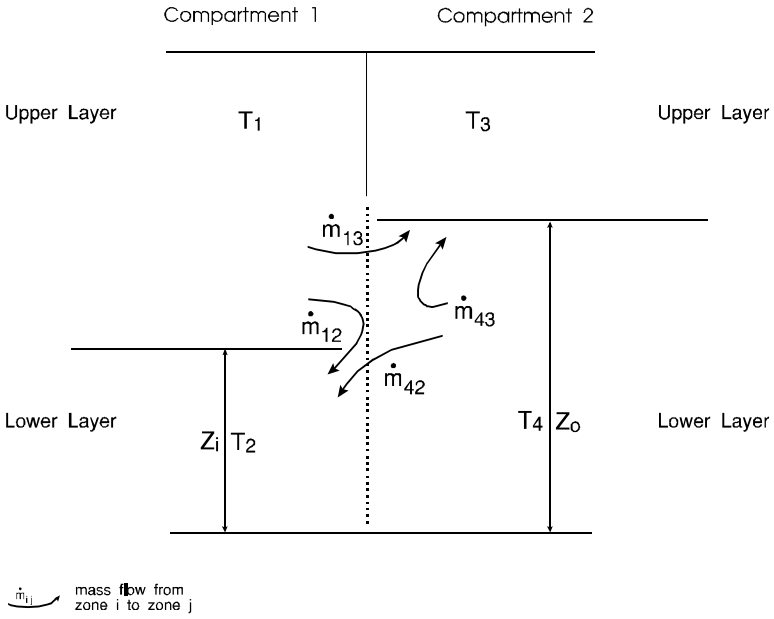


Figure 5. Example of a possible flow pattern and layer numbering convention.

The other type of mixing is much like an inverse plume and causes contamination of the lower layer. It occurs when there is flow of the type $\dot{m}_{42} > 0$. The shear flow causes vortex shedding into the lower layer and thus some of the particulates end up in the lower layer. The actual amount of mass or energy transferred is usually not large, but its effect can be large. For example, even minute amounts of carbon can change the radiative properties of the gas layer, from negligible to something finite. It changes the rate of radiation absorption by orders of magnitude which invalidates the notion of a diathermous lower layer. This term is predicated on the Kelvin-Helmholz flow instability and requires shear flow between two separate fluids. The mixing is enhanced for greater density differences between the two layers. However, the amount of mixing has never been well characterized. Quintiere et al. discuss this phenomena for the case of crib fires in a single room, but their correlation does not yield good agreement with experimental data in the general case [32]. In the CFAST model, it is assumed that the incoming cold plume behaves like the inverse of the usual door jet between adjacent hot layers; thus we have a descending plume. It is possible that the entrainment is overestimated in this case, since buoyancy, which is the driving force, is not nearly as strong as for the usually upright plume.

Implementation (HFLOW, VENT, FLOGO1, ENTRAIN): The physical interface routine HFLOW takes the mass flow and temperature for each vertical vent calculated by the physical routine VENT and calculates mass and enthalpy flow into each layer in a form appropriate for the physical model. For each vertical vent, the following calculation is performed:

1. For each vent, the routine GETVAR obtains the environment (temperature, pressure, density, species concentrations) for the two compartments connected by the vent.

2. The routine VENT divides the vent into slabs and determines the magnitude and direction of the mass and enthalpy flow in each slab.

3. The routine FLOGO1 implements a set of rules for depositing the flow from each slab into an appropriate layer in the destination compartment and removing the flow from the source room for each slab.

4. Mixing at the vent is calculated by the routine ENTRAIN.

3.3.2 Vertical Flow Through Horizontal Vents

Flow through a ceiling or floor vent can be somewhat more complicated than through door or window vents. The simplest form is uni-directional flow, driven solely by a pressure difference. This is analogous to flow in the horizontal direction driven by a piston effect of expanding gases. Once again, it can be calculated based on the Bernoulli equation, and presents little difficulty. However, in general we must deal with more complex situations that must be modeled in order to have a proper understanding of smoke movement. The first is an occurrence of puffing. When a fire exists in a compartment in which there is only a hole in the ceiling, the fire will burn until the oxygen has been depleted, pushing gas out the hole. Eventually the fire will die down. At this point ambient air will rush back in, enable combustion to increase, and the process will be repeated. Combustion is thus tightly coupled to the flow. The other case is exchange flow which occurs when the fluid configuration across the vent is unstable (such as a hotter gas layer underneath a cooler gas layer). Both of these pressure regimes require a calculation of the onset of the flow reversal mechanism.

Theory: Normally a non-zero cross vent pressure difference tends to drive unidirectional flow from the higher to the lower pressure side. An unstable fluid density configuration occurs when the pressure alone would dictate stable stratification, but the fluid densities are reversed. That is, the hotter gas is underneath the cooler gas. Flow induced by such an unstable fluid density configuration tends to lead to bi-directional flow, with the fluid in the lower compartment rising into the upper compartment. This situation might arise in a real fire if the room of origin suddenly had a hole punched in the ceiling. We make no pretense of being able to do this instability calculation analytically. We use Coopers's algorithm [33] for computing mass flow through ceiling and floor vents. It is based on correlations to model the unsteady component of the flow. What is surprising is that we can find a correlation at all for such a complex phenomenon. There are two components to the flow. The first is a net flow dictated by a pressure difference. The second is an exchange flow based on the relative densities of the gases. The overall flow is given by [33]

$$\dot{m} = C f(\gamma,\varepsilon)\left(\frac{\delta P}{\bar{\rho}}\right)^{1/2} A_v \tag{53}$$

where

$$C = 0.68 + 0.17\varepsilon, \tag{54}$$

$$\varepsilon = \frac{\delta P}{P}, \tag{55}$$

and f is a weak function of both γ and ε. In the situation where we have an instability, we use Cooper's correlations. The algorithm for this exchange flow is given by

$$\dot{m}_{ex} = 0.1\left(\frac{g\,\delta\rho\,A_v^{5/2}}{\rho_{av}}\right)\left(1.0 - \frac{2\,A_v^2\,\delta\rho}{S^2\,g\,\delta\rho\,D^5}\right) \tag{56}$$

where

$$D = 2\sqrt{\frac{A_v}{\pi}} \tag{57}$$

and S is 0.754 or 0.942 for round or square openings, respectively.

A simple example of the effect of this exchange flow can be shown with the following example. Consider two closed compartments, each 10 m in height, one on top of the other, connected by a one meter diameter round hole. Given hydrostatic equilibrium, there will be no flow between the compartments. Varying the pressure and density of the gas in the lower compartment very slightly from initial values of 1.2 kg/m^3 and 117.6 N/m^2 respectively, we calculate the flow between the compartments, as shown in Figure 6.

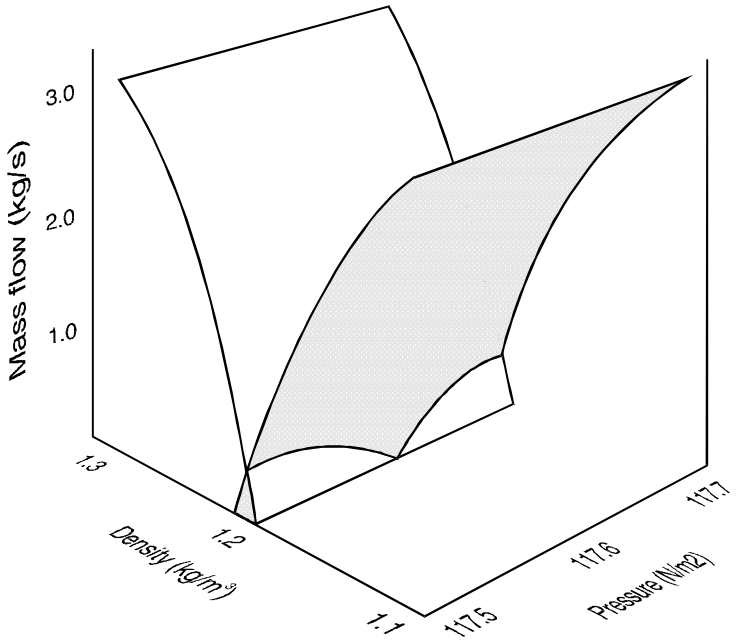


Figure 6. An example of vertical flow between two compartments.

Implementation (VFLOW and VENTCF): The physical interface routine VFLOW takes the mass flow and temperature for each horizontal vent calculated by the physical routine VENTCF and calculates mass and enthalpy flow into each layer in a form appropriate for the physical model. For each horizontal vent, the following calculation is performed:

1. For each vent, the routine VFLOW determines the top and bottom compartments. It then calls the physical routine VENTCF to get the amount of mass, the temperature of that mass, and the layer the mass flows into for each room.

2. VFLOW then takes the mass into each room multiplies it by the temperature and the specific heat of the gas to get the enthalpy. To determine which layer the flow goes into the temperature of the mass is compared to the temperature of the upper layer of the room getting the mass. If the temperature is cooler than the upper layer then the mass goes to the lower layer, otherwise it goes to the upper. The enthalpy is then subtracted from the appropriate layer in the room the mass is coming from. It is assumed that mass flowing into the upper room comes from the layer nearest the vent in the lower room, the upper layer. The same is true in the reverse case, the mass from the upper room to the lower room comes from the lower layer in the upper room.

3. The routine VFLOW then determines the correct amount of each species to put into the receiving layer and subtracts the amount from the contributing layer. This is done by using the mass fraction of each of the species in the contributing layer times the total mass moved.

3.3.3 Forced Flow

The model for mechanical ventilation is based on the theory of networks and is based on the model developed by Klote [34]. This is a simplified form of Kirchoff's law which says that flow into a node must be balanced by flow out of the node. There is a close analog to electrical networks for which the flow consists of electrons. In the case of ventilation, the flow is formed by molecules of air. The conservation equation differs slightly from that of an electrical system, but the basic ideas carry over. For the former case, we have:

$$\text{voltage} = \text{current} \times \text{resistance} .$$

In the present case we have

$$\text{pressure change} = \text{mass flow} \times \text{resistance} .$$

So the application of network theory is used, although the circuit laws are slightly different. In practice, as with the electrical analog, one solves the problem by summing all of the equations for the nodes, and requires that the mass be conserved at each node. Thus we turn the equation around and put it into the form

$$\text{mass flow} = \text{conductance} \times (\text{pressure drop across a resistance})^{1/2} .$$

For each node, this flow must sum to zero. There are several assumptions which are made in computing this flow in ducts, fans, elbow, *etc*. First, we assume unidirectional flow. Given the usual size of ducts, and the nominal presence of fans, this is quite reasonable. Also, the particular implementation used here [34] does not allow for reverse flow in the duct system. The difficulty lies in describing how a fan behaves in such a case.

Theory: Given that we can describe mass flow in terms of pressure differences and conductance, the conservation equation for each node is

$$\sum_j \dot{m}_{i,j} = 0. \tag{58}$$

The index "*j*" is a summation over connections to a node, and there is an equation "*i*" for each node. The remaining problem is to specify the boundary conditions. At each connection to a compartment, the pressure is specified. Then, given that flow at each connection is unidirectional (at a given instant of time, the flow is either all into or all out of a given connection), the mass and enthalpy flow into or out of a room can be calculated explicitly. Thus we end up with a set of equations of the form

$$\begin{aligned} f_1(P_1,P_2, \ldots) &= 0 \\ &\vdots \\ f_i(P_1,P_2, \ldots) &= 0 \\ &\vdots \\ f_n(P_1,P_2, \ldots) &= 0. \end{aligned} \tag{59}$$

This is an algebraic set of equations that is solved simultaneously with the equations for flow in the compartments.

The equations describe the relationship between the pressure drop across a duct, the resistance of a duct, and the mass flow. The pressure can be changed by conditions in a compartment, or a fan in line in the duct system. Resistance arises from the finite size of ducts, roughness on surfaces, bends and joints. To carry the electrical analog a little further, fans act like constant voltage sources. The analogy breaks down, however, in that the voltage, current and resistance are related by the square of the current, rather than being linearly proportional. Since we are using the current form of the conservation equation to balance the system, the flow can be recast in terms of a conductance

$$\dot{m} = G\sqrt{\triangle P}. \tag{60}$$

The conductance can be expressed generally as

$$G = \sqrt{\frac{2\rho}{C_0}}\, A_0 \tag{61}$$

where C_0 is the flow coefficient, and A_0 is the area of the inlet, outlet, duct, contraction or expansion joint, coil, damper, bend, filter, and so on. Their values for the most common of these items are tabulated in the ASHRAE Handbook [35].

Ducts are long pipes through which gases can flow. They have been studied much more extensively than other types of connections. For this reason, eq (61) can be put into a form which allows one to characterize the conductance in more detail, depending on the type of duct (e.g., oval, round, or square). The form derives from the Darcy equation and is

$$G = \sqrt{\frac{FL}{2\rho D_e A_0^{\,2}}}\,, \tag{62}$$

where F is the friction factor and can be calculated from

$$\frac{1}{\sqrt{F}} = -2\,\log\left(\frac{\epsilon}{3.7\,D_e} + \frac{2.51}{R_e\sqrt{F}}\right). \tag{63}$$

For each node in the system, one has an entry of the form of eq (61).

The temperature for each duct *d* is determined using the following differential equation

accumulated heat = (heat in - heat out) - convective losses through duct walls

$$c_v \rho_d V_d \frac{dT_d}{dt} = c_p m_d \left(T_{in} - T_{out}\right) - h_d A_d \left(T_d - T_{amb}\right) \tag{64}$$

where c_v, c_p are the specific heats at constant volume, pressure; V_d is the duct volume, ρ_d is the duct gas density, dT_d/dt is the time rate of change of the duct gas temperature, m_d is the mass flow rate,T_{in} and T_{out} are the gas temperatures going into and out of the duct, c_d, A_d are the convective heat transfer coefficient and surface area for duct d and T_{amb} is the ambient temperature. The first term on the right hand side of eq (64) represents the net gain of energy due to gas transported into or out of the duct. The second term represents heat transferred to the duct walls due to convection. In version 1.6, the loss coefficient is set to zero. We retain the form for future work. The differential and algebraic (DAE) solver used by CFAST solves eq (64) exactly as written. A normal ordinary differential equation solver would require that this equation be solved for dT/dt. By writing it this way, the duct volumes can be zero which is the case for fans.

The mechanical ventilation system is partitioned into one or more independent systems. Differential equations for species for each of these systems are derived by lumping all ducts in a system into one pseudo tank. The equations for each tank are solved using time splitting similar to how the gas layer species are computed.

This set of equations is then solved at each time step. Previously the mechanical ventilation computations in CFAST were performed as a side calculation using time splitting. This could cause problems since time-splitting methods require that the split phenomenon (the pressures and temperatures in this case) change slowly compared to other phenomenon such as room pressures, layer heights etc. The pressures at each internal node and the temperatures in each branch (duct, fan) are now determined explicitly by the solver once again using conservation of mass and energy.

Implementation (MVENT, HVMFLO and HVSFLO): The physical interface routine MVENT takes the mass flow and temperature for each mechanical ventilation system calculated by the physical routine HVMFLO (mass) and HVSFLO (temperature) and calculates mass and enthalpy flow into each layer of the compartments connected to the system in a form appropriate for the solver interface. In addition, it calculates the pressure and temperature at each node in the system to form a set of linear equations for solution by the DAE solver used in CFAST. For each, the following calculation is performed:

1. For each connection between a mechanical ventilation system and a compartment, the routine HVFREX is called to obtain compartment pressures, temperatures and concentrations at the connections.

2. For each duct and fan in a mechanical ventilation system, the routines HVMFLO and HVFAN calculate the mass flow through the duct or fan.

3. For each connection between a mechanical ventilation system and a compartment, the routines HVSFLO and HVTOEX are called to compute the differential equation for temperature and species in the mechanical ventilation system.

3.4 Heat Transfer

3.4.1 Radiation

Objects such as walls, gases and fires radiate as well as absorb radiation. Each object has its own properties, such as temperature and emissivity. As we are solving the enthalpy equation for the gas temperature, the

primary focus is in finding out how much enthalpy is gained or lost by the gas layers due to radiation. To calculate the radiation absorbed in a zone, a heat balance must be done which includes all surfaces which radiate to and absorb radiation from a zone. The form of the terms which contribute heat to an absorbing layer are the same for all layers. Essentially we assume that all zones in these models are similar so we can discuss them in terms of a general layer contribution. For this calculation to be done in a time commensurate with the other sources, some approximations are necessary.

Radiation can leave a layer by going to another layer, by going to the walls, by exiting through a vent, by heating an object, or by changing the pyrolysis rate of the fuel source. Similarly, a layer can be heated by absorption of radiation from these surfaces and objects as well as from the fire itself. The formalism which we employ for the geometry and view factor calculation is that of Siegel and Howell [37]. Although the radiation could be done with a great deal of generality, we have assumed that the zones and surfaces radiate and absorb like a grey body.

Radiation is an important mechanism for heat exchange in compartments subject to fires. It is important in the present application because it can affect the temperature distribution within a compartment, and thus the buoyancy forces. In the present implementation the fire is assumed to be a point source; it is assumed that plumes do not radiate. We use a simplified geometrical equivalent of the compartment in order to calculate the radiative transfer between the ceiling, floor and layer(s). The original paper which described FAST pointed out that there was an inconsistency in the interaction between the walls and the radiation from and to the gas layers. This modification fixes that problem. A radiative heat transfer calculation could easily dominate the computation in any fire model. This is because radiation exchange is a global phenomena. Each portion of an enclosure interacts radiatively with every other portion that it "sees." Therefore, it is important to construct algorithms for radiative heat transfer that are both accurate and efficient [36].

This is a "next step" algorithm for computing radiative heat transfer between the bounding surfaces of a compartment containing upper and lower layer gasses and point source fires. The two-wall radiation model used has been enhanced to treat lower layer heating and to treat radiative heat exchange with the upper and lower walls independently of the floor and ceiling. We refer to this as the four wall model.

The original radiation algorithm used the extended floor and ceiling concept for computing radiative heat exchange. For the purposes of this calculation, the room is assumed to consist of two wall segments: an extended ceiling and an extended floor. The extended ceiling consisted of the ceiling plus the upper wall segments. Similarly, the extended floor consisted of the floor plus the lower wall segments. The upper layer was modeled as a sphere equal in volume to the volume of the upper layer. Radiative heat transfer to and from the lower layer was ignored. This algorithm is inconsistent with the way heat conduction is handled, since we solve up to four heat conduction problems for each room: the ceiling, the upper wall, the lower wall and the floor. The purpose of the new radiation algorithm then is to enhance the radiative module to allow the ceiling, the upper wall segments, the lower wall segments and the floor to transfer radiant heat independently and consistently.

Theory: The four wall algorithm for computing radiative heat exchange is based upon the equations developed in Siegel and Howell [37] which in turn is based on the work of Hottel [38]. Siegel and Howell model an enclosure with N wall segments and an interior gas. A radiation algorithm for a two layer zone fire model requires treatment of an enclosure with two uniform gases. Hottel and Cohen [39] developed a method where the enclosure is divided into a number of wall and gas volume elements. An energy balance is written for each element. Each balance includes interactions with all other elements. Treatment of the fire and the interaction of the fire and gas layers with the walls is based upon the work of Yamada and Cooper [40]. They

model fires as point heat sources radiating uniformly in all directions and use the Lambert-Beer law to model the interaction between heat emitting elements (fires, walls, gas layers) and the gas layers. The original formulation is for an N-wall configuration. Although this approach would allow arbitrary specification of compartment surfaces (glass window walls, for example), the computational requirements are significant.

Even the more modest approach of a four wall configuration for computing radiative heat transfer is more sophisticated than was used previously. By implementing a four wall rather than an N wall model, significant algorithmic speed increases were achieved. This was done by exploiting the simpler structure of the four wall problem.

The radiation exchange at the k'th surface is shown schematically in Figure 7. For each wall segment k from 1 to N we must find a net heat flux, $\Delta q_k''$, such that

$$A_k \sigma_k T_k^4 q_k^{in} + (1-\epsilon_k) q_k^{in} = q_k^{in} + A_k \Delta q_k'' \quad (k = 1,..N). \tag{65}$$

Radiation exchange at each wall segment has emitted, reflected, incoming and net radiation terms. Equation (65) then represents a system of linear equations that must be solved for $\Delta q''$ to determine the net fluxes given off by each surface. The setup and solution of this linear system is the bulk of the work required to implement the net radiation method of Siegel and Howell. Equation (66) derived by Siegel and Howell [37] and listed there as eqs 17 to 20, is called the **net radiation equation**,

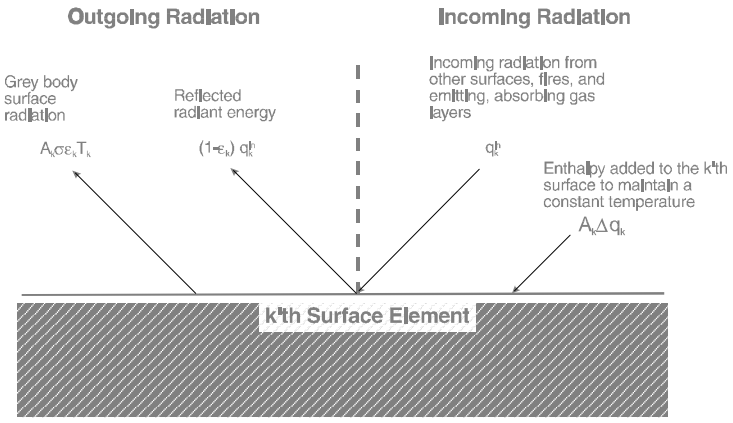


Figure 7. Radiation exchange in a two-zone fire model.

$$\frac{\Delta q_k''}{\epsilon_k} - \sum_{J=1}^{N} \frac{1-\epsilon}{\epsilon_j} \Delta q_j'' F_{k-j} \tau_{j-k} = \sigma T_k^4 - \sum_{j=1}^{N} \sigma T_j^4 F_{k-j} \tau_{j-k} - \frac{c_k}{A_k}. \tag{66}$$

where σ is the Stefan-Boltzman constant, ϵ_k is the emissivity of the k'th wall segment, T_k is the temperature of the k'th wall segment, F_{k-j} a configuration factor, and τ is a transmissivity factor. This latter is the fraction of energy passing unimpeded through a gas along a path from surface j to k. The parameters c_k represent the various sources of heat, namely the fire itself and the gas layers. In the form shown, the view factor of the k'th element is included in the parameter c.

The actual implementation uses a slightly modified form of eq (66), namely

$$\Delta \hat{q}_k'' - \sum_{j=1}^{N} (1-\epsilon_j) \Delta \hat{q}_j'' F_{k-j} \tau_{j-k} = \sigma T_j^4 - \sum_{j=1}^{N} \sigma T_j^4 F_{k-j} \tau_{j-k} - \frac{c_k}{A_k}, \text{ where} \tag{67}$$

$$\Delta q_k'' = \epsilon_k \Delta \hat{q}_k''. \qquad (68)$$

There are two reasons for solving eq (67) rather than eq (66). First, since ε_k does not occur in the denominator, radiation exchange can be calculated when some of the wall segments have zero emissivity. Second and more importantly, the matrix corresponding to the linear system of eq (68) is diagonally dominant [36]. Iterative algorithms can be used to solve such systems more efficiently than direct methods such as Gaussian elimination. The more diagonally dominant a matrix (the closer the emissivities are to unity), the quicker the convergence when using iterative methods. Typical values of the emissivity for walls subject to a fire environment are in the range of $0.85 < \varepsilon < 0.95$, so this is a reasonable approximation. The computation of, $F_{k\text{-}j}$, $\tau_{j\text{-}k}$ and c_k is discussed by Forney [36]. It is shown how it is possible to use the symmetries present in the four wall segment problem to minimize the number of direct configuration factor calculations required.

For rooms containing a fire, CFAST models the temperature of four wall segments independently. A two wall model for radiation exchange can break down when the temperatures of the ceiling and upper walls differ significantly. This typically happens in the room of fire origin when different wall materials are used as boundaries for the ceiling, walls and floor. To demonstrate this consider the following example.

To simplify the comparison between the two and four wall segment models, assume that the wall segments are black bodies (the emissivities of all wall segments are one) and the gas layers are transparent (the gas absorptivities are zero). This is legitimate since for this example we are only interested in comparing how a two wall and a four wall radiation algorithm transfer heat to the wall segments. Let the room dimensions be 4×4×4 [m], the temperature of the floor and the lower and upper walls be 300 K. Let the ceiling temperature vary from 300 to 600 K.

Figure 8 shows a plot of the heat flux striking the ceiling and upper wall as a function of the ceiling temperature. The two wall model predicts that the extended ceiling (a surface formed by combining the ceiling and upper wall into one wall segment) cools, while the four wall model predicts that the ceiling cools and the upper wall warms. The four-wall model moderates temperature differences that may exist between the ceiling and upper wall (or floor and lower wall) by allowing heat transfer to occur between the ceiling and upper wall. The two wall model is unable to predict heat transfer between the ceiling and the upper wall since it models them both as one wall segment.

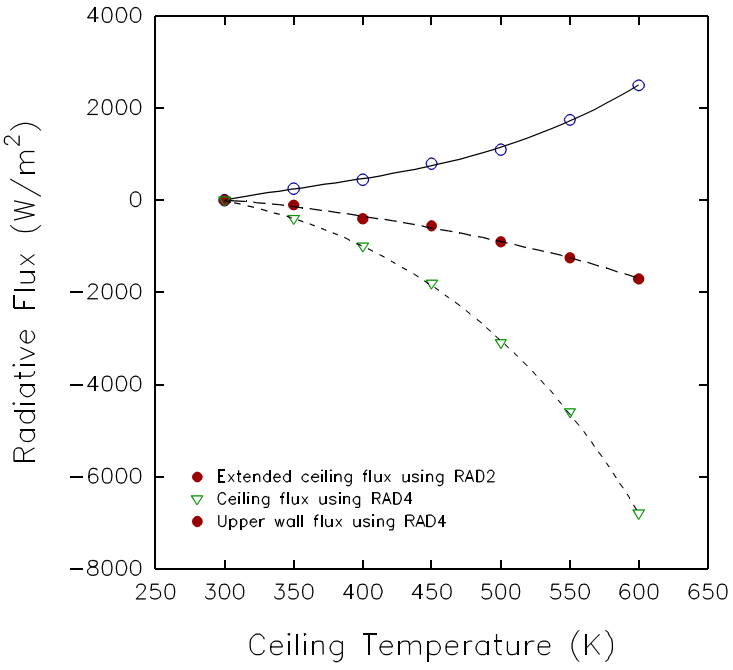


Figure 8. An example of two-wall and four-wall calculations for radiation exchange on a ceiling and wall surface.

Implementation (RDHEAT, RAD4, and RAD2): RDHEAT is the physical interface routine between CFAST and the physical routines RAD4 and RAD2

for four-wall and two-wall radiation calculations, respectively. RDHEAT returns the flux to all surfaces from each layer and each fire due to radiation. It also returns the change in enthalpy in gas layers for rooms containing a fire:

1. RDHEAT works room by room in its calculations. First for a given room, each of the four surfaces are checked to see if the surface is a conducting surface or not.

2. For conducting surfaces the wall temperature supplied by the solver, DASSL, and the material emissivity are used. If the surface is "off" (assumed to be non-conducting, then the temperature of the adjacent layer is used and the emissivity is set to zero.

3. If the room being considered contains one or more fires, the routine RAD4 which is a four wall radiation algorithm is used. If there are no fires in the room then RAD2, a two wall radiation algorithm, is used. Both routines return the net flux to each surface and the net change in enthalpy of each layer in the room.

3.4.2 Convection

Convection is one of the mechanisms by which the gas layers lose or gain energy to walls, objects or through openings. Conduction is a process which is intimately associated with convection; but as it does not show up directly as a term for heat gain or loss, it will be discussed separately. Convective heating describes the energy transfer between solids and gases. The enthalpy transfer associated with flow through openings will be discussed in the section on flow through vents.

Theory: Convective heat flow is enthalpy transfer across a thin boundary layer. The thickness of this layer is determined by the temperature difference between the gas zone and the wall or object being heated [41]. We can write the heat flux term as

$$\dot{Q}_c = h_c(T_g - T_w)A_w \tag{69}$$

where the transfer coefficient (assuming natural convection) can be written as

$$\begin{aligned} h_c &= \frac{\kappa}{l} C_o (Gr\ Pr)^{1/3} \\ Gr &= \frac{g l^3 |T_g - T_w|}{\upsilon^2 T_g} \\ \kappa &= 2.72\times10^{-4} \left(\frac{T_g + T_w}{2} \right)^{4/5} \\ \upsilon &= 7.18\times10^{-10} \left(\frac{T_g + T_w}{2} \right)^{7/4} \end{aligned} \tag{70}$$

and T_g and T_w are the temperatures of the gas layer and the wall respectively, A_w is the area of surfaces in contact with the zone, Pr is the Prandtl number (0.72), l is a characteristic length scale $\approx (A_w)^{1/2}$, and C_o is a coefficient which depends on orientation [41].

For the cases of interest we use the coefficients shown below. The coefficients for horizontal surfaces apply to a slab over a zone, such as ceiling surfaces. For a floor, the conditions (T_g and T_w) are reversed. For the outside boundary, the condition is reversed, at least for the ceiling and floor. Physically, outside a compartment, the ceiling of a compartment will behave as if it were the floor of a compartment over it, and similarly for the floor of a compartment. Thus, we use the floor boundary coefficient for the outside boundary of the ceiling and the ceiling coefficient for the outside boundary of a compartment floor. For vertical boundaries, the coefficient remains the same on the interior and exterior.

Orientation	Coefficient[C_o]	Condition
Vertical	0.130	all
Horizontal	0.210	$T_g > T_w$
Horizontal	0.012	$T_g < T_w$

These coefficients are for turbulent boundary layer flow. They overestimate the heat transfer which can occur in a quiescent compartment.

The boundary condition which connects the interior of the wall to the zone is fairly straightforward. This convective heating generates a flux from the gas layer which becomes a derivative boundary condition for the conduction algorithm. A similar boundary condition must be applied on the exterior of the walls. The assumption made is that the exterior portion of a wall is truly facing the ambient. This precludes a fire in one compartment heating a connected compartment through conduction. The omission is due to the difficulty of specifying how compartment walls are connected and not to the difficulty of specifying the boundary conditions or solving the equations. So the boundary condition for the exterior of a wall is similar to the interior, except that the exterior surface is assumed to be convecting and radiating to the ambient. With this caveat in mind, we can use the convection routine to calculate the boundary condition for the exterior wall also.

The current model allows for a ceiling, floor and two walls. Actually the two walls are the same material, but a separate temperature profile is maintained for the wall in contact with the upper and lower zones respectively. Therefore we have four components for convective heat transfer.

Implementation (CVHEAT and CONVEC): CVHEAT is the physical interface routine between CFAST and the physical routine CONVEC. CVHEAT returns to CFAST the convective heat transfer to each surface in a compartment and the change in enthalpy for each layer.

1. For each compartment in the simulation, the routine CVHEAT determines the areas of the upper wall and lower wall; the ceiling and floor are of course constant.

2. For a conducting surface, a check is made to see if both the ceiling jet is being calculated and at least one fire is in the room. The ceiling jet algorithm supersedes the convection algorithm for calculating flux to a surface in a fire room.

3. If the ceiling jet calculation is included, CVHEAT returns a zero for the flux to the surfaces. Otherwise CVHEAT calls the physical routine CONVEC to find the flux to each of the surfaces. The enthalpy that goes to the walls, ceiling, and floors is subtracted from the appropriate layer to conserve enthalpy. This is done by multiplying the flux to the surface by the area of the surface and subtracting it from the layer.

3.4.3 Conduction

Theory: Conduction of heat through solids is not a source term in the sense discussed earlier. That is, loss or gain of energy from solids occurs by convective heating, which in turn is influenced by subsequent gain or loss through the solids. However, as much of the net heat loss from a compartment occurs through loss to the walls and heating of interior objects and thus provides the boundary conditions for the other source terms discussed above.

The equation which governs the heat transfer in solids is

$$\frac{\partial T}{\partial t} = \frac{k}{\rho c}\nabla^2 T \tag{71}$$

and is a linear parabolic equation. As such it must be solved by a different technique than is used for the ordinary differential equations which describe mass and enthalpy flux. The equation is linear only if the coefficients k, ρ and c are independent of temperature throughout the material. This may not be the case, especially for some materials such as gypsum for which the value of k may vary by a factor of two or more. However, to the accuracy that we know most of the thermal properties, it is a reasonable approximation. Procedures for solving 1-d heat conduction problems are well known. For finite difference methods such as backward difference (fully implicit), forward difference (fully explicit) or Crank-Nicolson, see [42]. For finite element methods see [43].

In earlier versions of the model, time splitting was employed to reduce the computation time for the conduction calculation. Time splitting makes the assumption that two or more phenomena change over significantly different time scales. For example, in the zone fire modeling case, it can often be assumed that the characteristic time scale for wall segment temperature profiles is much longer than that for the gas solution variables. Suppose that the gas solution variables and the wall segment temperature profiles are known at time t. If the characteristic time scale for wall segment temperature profiles is δt, then wall segment temperature profiles would be solved over the time interval $(t, t+\delta t)$. This time interval would then be further subdivided in order to solve for the shorter time scale phenomena. The longer time interval is often called the outer time step and the shorter interval is called the inner time step.

The method of time splitting does not work well when the time scales are close, which can occur when wall materials are thin and/or highly conductive. Time splitting is also difficult to implement efficiently since it is not clear what time step sizes should be used. A time stepsize chosen too small will result in inefficiency and time step size chosen too large will result in unnecessarily inaccurate answers. The former can easily occur as a fire simulation approaches steady state. In this case, solution variables do not change much and inefficiency occurs because of restrictive time stepsize selections.

To advance the solution for the wall temperature profile, a finite difference approach [44] was used. A graded (non-uniform) mesh with n_x breakpoints was introduced for the spatial variable x. The second spatial derivative in the heat equation was replaced by a second divided (finite) difference approximation. This produces a system of n_{x-2} ODE's for the n_{x-2} unknown temperatures at the interior breakpoints. The temperatures at the interior boundary is supplied by the differential equation solver. The exterior boundary conditions (constant flux, insulated, or constant temperature) is specified in the configuration of CFAST. This system was solved by one step of the backward Euler method. Crank-Nicholson was also tried but

produced spurious oscillations in the temperature profiles at the beginning of the simulation; backward Euler does not suffer from this defect which is related to the non-uniform mesh being used. The solution at time $t + \delta t$ can be found by solving a tridiagonal system of linear equations. The temperature gradient at $x=0$ and time $t + \delta t$ was approximated by computing a derivative difference using the first two temperatures.

A graded mesh scheme was chosen to allow breakpoints to cluster near the interior and exterior wall segment surfaces. This is where the temperature gradients are the steepest. A breakpoint x_b was defined by $x_b=MIN(x_p,W/2)$,where $x_p = 2\left(\alpha t_{final}\right)^{\frac{1}{2}} \text{erfc}^{-1}(.05)$ and erfc^{-1} denotes the inverse of the complementary error function. The value x_p is the location in a semi-infinite wall where the temperature rise is 5 percent after t_{final} seconds and is sometimes called the penetration depth. Eighty percent of the breakpoints were placed on the interior side of x_b and the remaining 20 percent were placed on the exterior side.

In this version of the model, we use a new strategy for coupling the 1-d heat conduction problem with the ODE's for the gas solution variables based on the work of Moss and Forney [44]. This method couples the wall segment surface temperatures, rather than the entire wall segment temperature profile, with the gas solution variables by requiring that the wall segment surface temperature gradient, $\partial u(x,t)/\partial x$, and the incident heat flux (sum of convective and net radiative flux), q'' satisfy Fourier's law

$$q'' = -\kappa \frac{\partial u(x,t)}{\partial x} \tag{72}$$

at the wall boundary $x=0$ where K is the thermal conductivity of the wall material. This solution strategy requires a DAE solver that can simultaneously solve both differential (gas ODE's) and algebraic equations (Fourier's law). With this method, only one or two extra equations are required per wall segment (two if both the interior and exterior wall segment surface temperatures are computed). This solution strategy is more efficient than the method of lines since fewer equations need to be solved. Wall segment temperature profiles, however, still have to be stored so there is no decrease in storage requirements.

To illustrate the method, consider a one room case with one active wall. There will be four gas equations (pressure, upper layer volume, upper layer temperature, and lower layer temperature) and one wall temperature equation. Implementation of the gradient matching method requires that storage be allocated for the temperature profile at the previous time, t, and at the next time, $t + \delta t$. Given the profile at time t and values for the five unknowns at time $t + \delta t$ (initial guess by the solver), the temperature profile is advanced from time t to time $t + \delta t$. The temperature profile gradient at $x = 0$ is computed followed by the residuals for the five equations. The DAE solver adjusts the solution variables and the time step until the residuals for all the equations are below an error tolerance. Once the solver has completed the step, the array storing the temperature profile for the previous time is updated, and the DAE solver is ready to take its next step.

One limitation of our implementation of conduction is that it serves only as a loss term for enthalpy. Heat lost from a compartment by conduction is assumed to be lost to the outside ambient. In reality, compartments adjacent to the room which contains the fire can be heated, possibly catastrophically, by conducted energy not accounted for in the model. Although solving the conduction equations for this situation is not difficult, the geometrical specification is. For this reason, we have chosen to assume that the outside of a boundary is always the ambient. A means to connect compartments physically so that heat can be transported by conduction is under active study.

Implementation (CNHEAT and CNDUCT): CNHEAT is the physical interface routine between CFAST and the physical routine CNDUCT. CNHEAT returns to CFAST the conductive heat transfer through each surface in a compartment. For each surface, the routine CNDUCT calculates the conductive heat transfer through the surface material.

3.4.4 Ceiling Jet

Theory: Relatively early in the development of a fire, fire-driven ceiling jets and gas-to-ceiling convective heat transfer can play a significant role in room-to-room smoke spread and in the response of near-ceiling mounted detection hardware. Cooper [45] details a model and computer algorithm to predict the instantaneous rate of convective heat transfer from fire plume gases to the overhead ceiling surface in a room of fire origin. The room is assumed to be a rectangular parallelopiped and, at times of interest, ceiling temperatures are simulated as being uniform. Also presented is an estimate of the convective heat transfer due to ceiling-jet driven wall flows. The effect on the heat transfer of the location of the fire within the room is taken into account. This algorithm has been incorporated into the CFAST model. In this section, we provide an overview of the model. For complete details, we refer the reader to reference [45].

A schematic of a fire, fire plume, and ceiling jet is shown in Figure 9. The buoyant fire plume rises from the height Z_{fire} toward the ceiling. When the fire is below the layer interface, its mass and enthalpy flow are assumed to be deposited into the upper layer at height Z_{layer}. Having penetrated the interface, a portion of the plume typically continues to rise toward the ceiling. As it impinges on the ceiling surface, the plume gases turn and form a relatively high temperature, high velocity, turbulent ceiling jet which flows radially outward along the ceiling and transfers heat to the relatively cool ceiling surface. The convective heat transfer rate is a strong function of the radial distance from the point of impingement, reducing rapidly with increasing radius. Eventually, the relatively high temperature ceiling jet is blocked by the relatively cool wall surfaces [46]. The ceiling jet then turns downward and outward in a complicated flow along the vertical wall surfaces [47], [48]. The descent of the wall flows and the heat transfer from them are eventually stopped by upward buoyant forces. They are then buoyed back upward and mix with the upper layer.

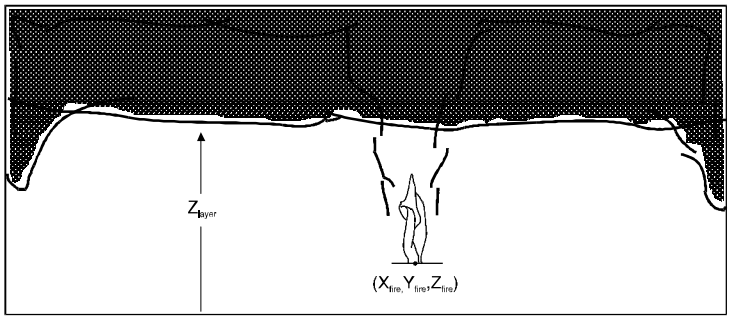


Figure 9. Convective heat transfer to ceiling and wall surfaces via the ceiling jet.

The average convective heat flux from the ceiling jet gases to the ceiling surface, Q_{ceil}, can be expressed in integral form as

$$\dot{Q}_{ceil} = \int_0^{X_{wall}} \int_0^{Y_{wall}} \dot{q}''_{ceil}(x,y)\, dxdy \tag{73}$$

The instantaneous convective heat flux, $\dot{q}''_{ceil}(X,Y)$ can be determined as derived by Cooper [45]:

$$\dot{q}''_{ceil}(x,y) = h_l \left(T_{ad} - T_{ceil}\right) \tag{74}$$

where T_{ad}, a characteristic ceiling jet temperature, is the temperature that would be measured adjacent to an adiabatic lower ceiling surface, and h_l is a heat transfer coefficient. h_l and T_{ad} are given by

$$\frac{h_l}{\tilde{h}} = \begin{cases} 8.82\,Re_H^{-1/2} Pr^{-2/3}\left(1-\left(5-0.284Re_H^{2/5}\right)\dfrac{r}{H}\right) & 0 \le \dfrac{r}{H} < 0.2 \\ 0.283 Re_H^{0.3} Pr^{-2/3}\left(\dfrac{r}{H}\right)^{-1.2} \dfrac{\dfrac{r}{H}-0.0771}{\dfrac{r}{H}+0.279} & 0.2 \le \dfrac{r}{H} \end{cases} \tag{75}$$

$$\frac{T_{ad}-T_u}{T_u Q_H^{*2/3}} = \begin{cases} 10.22 - 14.9\dfrac{r}{H} & 0 \le \dfrac{r}{H} < 0.2 \\ 8.39 f\left(\frac{r}{H}\right) & 0.2 \le \dfrac{r}{H} \end{cases} \tag{76}$$

where

$$f\left(\frac{r}{H}\right) = \frac{1 - 1.10\left(\dfrac{r}{H}\right)^{0.8} + 0.808\left(\dfrac{r}{H}\right)^{1.6}}{1 - 1.10\left(\dfrac{r}{H}\right)^{0.8} + 2.20\left(\dfrac{r}{H}\right)^{1.6} + 0.690\left(\dfrac{r}{H}\right)^{2.4}} \tag{77}$$

$$r = \left((X - X_{fire})^2 + (Y - Y_{fire})^2\right)^{\frac{1}{2}} \tag{78}$$

$${}_u C_p g^{1/2} H^{1/2} \dot{Q}_H^{*1/3}; \qquad Re_H = \frac{g^{1/2} H^{3/2} \dot{Q}_H^{*1/3}}{\nu_u}; \qquad \dot{Q}_H^* = \frac{\dot{Q}'}{\rho_u C_p T_u \sqrt{gH}} \tag{79}$$

$$\begin{cases} \dot{Q}_{fc}\dfrac{\sigma \dot{M}^*}{1+\sigma} & Z_{fire} < Z_{layer} < Z_{ceil} \\ \dot{Q}_{fc} & Z_{fire} \ge Z_{layer} \\ & Z_{layer} = Z_{ceil} \end{cases} \qquad \dot{M}^* = \begin{cases} 0 & -1 \\ \dfrac{1.04599\sigma + 0.360391\sigma^2}{1 + 1.37748\sigma + 0.360391\sigma^2} \end{cases} \tag{80}$$

$$\sigma = \frac{1 - \frac{T_u}{T_l} + C_T \dot{Q}_{EQ}^{*2/3}}{\frac{T_u}{T_l}}; \qquad C_t = 9.115 \tag{81}$$

$$\dot{Q}_{EQ}^* = \left(\frac{0.21\dot{Q}_{fc}}{C_p T_l \dot{m}_p}\right)^{3/2} \tag{82}$$

In the above, H is the distance from the (presumed) point source fire and the ceiling, X_{fire} and Y_{fire} are the position of the fire in the room, Pr is the Prandtl number (taken to be 0.7) and ν_u is the kinematic viscosity of the upper layer gas which is assumed to have the properties of air and can be estimated from $\nu_u = 0.04128(10^7)T_u^{5/2}/(T_u+110.4)$. Q_H^* and Q_{EQ}^* are dimensionless numbers and are measures of the strength of the plume at the ceiling and the layer interface, respectively.

When the ceiling jet is blocked by the wall surfaces, the rate of heat transfer to the surface increases. Reference [45] provides details of the calculation of wall surface area and convective heat flux for the wall surfaces.

Implementation (CJET and CEILHT): CJET is the physical interface routine between CFAST and the physical routine CEILHT. CJET calculates the convective heat transfer from the layers to the walls in rooms which contain one or more fires. Unlike the physical routine CVHEAT, CJET accounts for the spot heating of the ceiling and other surfaces due to an arbitrarily positioned fire:

1. For each compartment in the simulation, the routine CJET calculates the surface areas of the upper wall and lower wall. For surfaces that are included in the ceiling jet calculation in a room with at least one fire in the compartment, the effect of the ceiling jet for each fire is calculated independently.

2. For each fire, the routine CEILHT is called. On return from CEILHT the flux to the walls is summed for all the fires. After the inner loop is finished, a series of checks for each of the surfaces is done. If the surface is adiabatic, the ceiling jet is not to be calculated for that surface, or there are no fires in that compartment, the flux to the surface is set to zero. Otherwise flux is set to the value calculated by CEILHT.

3. The amount of energy lost to a surface is simply the flux to the surface times its area. To assure energy conservation, the energy going into a surface is subtracted from the adjacent gas layer.

CVHEAT and CJET are mutually exclusive. That is, one or the other is used to calculate the heat loss to a ceiling, depending on the model configuration.

3.5 Species Concentration and Deposition

3.5.1 Species Transport

Theory: The species transport in CFAST is really a matter of bookkeeping to track individual species mass as it is generated by a fire, transported through vents, or mixed between layers in a compartment. When the layers are initialized at the start of the simulation, they are set to ambient conditions. These are the initial temperature specified by the user, and 23 percent by mass (21 percent by volume) oxygen, 77 percent by mass (79 percent by volume) nitrogen, a mass concentration of water specified by the user as a relative humidity, and a zero concentration of all other species. As fuel is burned, the various species are produced in direct relation to the mass of fuel burned (this relation is the species yield specified by the user for the fuel burning). Since oxygen is consumed rather than produced by the burning, the "yield" of oxygen is negative, and is set internally to correspond to the amount of oxygen used to burn the fuel (within the constraint of available oxygen limits discussed in sec. 3.1.2).

Each unit mass of a species produced is carried in the flow to the various rooms and accumulates in the layers. The model keeps track of the mass of each species in each layer, and knows the volume of each layer as a function of time. The mass divided by the volume is the mass concentration, which along with the molecular weight gives the concentration in volume percent or ppm as appropriate.

Implementation: The species transport is integral to all of the physical interface routines in CFAST. Each physical interface routine is responsible for calculating the flow of total mass and each individual species into each layer of each compartment in a simulation.

3.5.2 HCl Deposition

Hydrogen chloride produced in a fire can produce a strong irritant reaction that can impair escape from the fire. It has been shown [49] that significant amounts of the substance can be removed by adsorption by surfaces which contact smoke. In our model, HCl production is treated in a manner similar to other species. However, an additional term is required to allow for deposition on, and subsequent absorption into, material surfaces.

Theory: The physical configuration that we are modeling is a gas layer adjacent to a surface (Figure 10). The gas layer is at some temperature T_g with a concomitant density of hydrogen chloride, ρ_{HCl}. The mass transport coefficient is calculated based on the Reynolds analogy with mass and heat transfer: that is, hydrogen chloride is mass being moved convectively in the boundary layer, and some of it simply sticks to the wall surface rather than completing the journey during the convective roll-up associated with eddy diffusion in the boundary layer. The boundary layer at the wall is then in equilibrium with the wall. The latter is a statistical process and is determined by evaporation from the wall and stickiness of the wall for HCl molecules. This latter is greatly influenced by the concentration of water in the gas, in the boundary layer and on the wall itself.

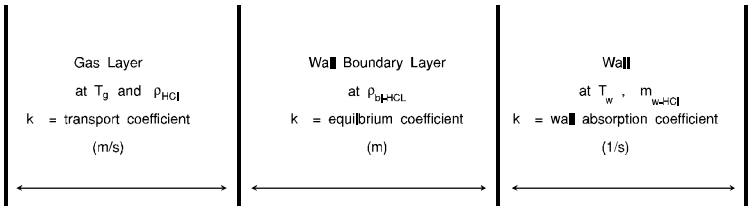


Figure 10. Schematic of hydrogen chloride deposition region.

The rate of addition of mass of hydrogen chloride to the gas layer is given by

$$\frac{d}{dt} m_{HCl} = source - k_c \times \left(\rho_{HCl} - \rho_{blHCl}\right) \times A_w \tag{83}$$

where source is the production rate from the burning object plus flow from other compartments.

For the wall concentration, the rate of addition is

$$\frac{d}{dt} d_{HCl,w} = k_c \times \left(\rho_{HCl} - \rho_{blHCl}\right) - k_s \times m_{HCl,w} \tag{84}$$

where the concentration in the boundary layer, ρ_{blHCl}, is related to the wall surface concentration by the equilibrium constant k_e,

$$\rho_{blHCl} = d_{HCl,w} / k_e \ . \tag{85}$$

We never actually solve for the concentration in the boundary layer, but it is available, as is a boundary layer temperature if it were of interest. The transfer coefficients are

$$k_c = \frac{\dot{q}}{\Delta T \; \rho_g \; c_p} \tag{86}$$

$$k_e = \frac{b_1 \; e^{1500/T_w}}{1 + b_2 \; e^{1500/T_w} \; \rho_{hcl}} \left(1 + \frac{b_5 \left(\rho_{H_2O} \right)^{b6}}{\left(\rho_{H_2O,sat} - \rho_{H_2O,g} \right)^{b7}} \right) \tag{87}$$

$$k_s = b_3 \; e^{-\left(\frac{b_4}{R \; T_W} \right)} \; . \tag{88}$$

The only values currently available [50] for these quantities are shown in Table 3. The "*b*" coefficients are parameters which are found by fitting experimental data to eqs (83) through (88). These coefficients reproduce the adsorption and absorption of HCl reasonably well. Note though that error bars for these coefficients have not been reported in the literature.

Surface	b_1 (m)	b_2 (m^3/kg)	b_3 (s^{-1})	b_4 (J/g-mol)	b_5 [a]	b_6 [c]	b_7 [c]
Painted Gypsum	0.0063	191.8	0.0587	7476.	193	1.021	0.431
PMMA	9.6×10^{-5}	0.0137	0.0205	7476.	29	1.0	0.431
Ceiling Tile	4.0×10^{-3}	0.0548	0.123	7476.	30[b]	1.0	0.431
Cement Block	1.8×10^{-2}	5.48	0.497	7476.	30[b]	1.0	0.431
Marinite®	1.9×10^{-2}	0.137	0.030	7476.	30[b]	1.0	0.431

a units of b_5 are $(m^3/kg)^{(b_7-b_6)}$
b very approximate value, insufficient data for high confidence value
c non-dimensional

Table 1. Transfer coefficients for HCl deposition

The experimental basis for poly(methyl methacrylate) and gypsum cover a sufficiently wide range of conditions that they should be usable in a variety of practical situations. The parameters for the other surfaces do not have much experimental backing, and so their use should be limited to comparison purposes.

Implementation (HCL and HCLTRAN): The physical interface routine, HCL, calculates the rates of addition of mass, enthalpy, and species into all layers in all room from the deposition of HCl onto wall surfaces in the compartments of a simulation. For each room, the following scheme is employed:

1. For each surface in each compartment of the simulation, the physical interface routine HCL determines the area of the surface, physical properties of the gas layer in contact with the surface (HCl concentration, water vapor concentration, temperature, and density).

2. The physical routine HCLTRAN is called to calculate the deposition rate of HCl onto the surface. The individual losses of the species HCl are accumulated for each room in the simulation by the routine HCL and the total mass and enthalpy in each layer are adjusted appropriately to insure conservation.

4 Structure of the Model

In this chapter, details of the implementation of the model are presented. These include

- an overview of the model formulation,
- the structure of the model including the major routines implementing the various physical phenomena included in the model,
- the organization of data initialization and data input used by the model,
- the structure of data used to formulate the differential equations solved by the model,
- a summary of the main control routines in the model that are used to control all input and output, initialize the model and solve the appropriate differential equation set for the problem to be solved, and
- guidelines for modifying the model to include new or enhanced physical phenomena.

4.1 Subroutine Structure

The calling sequence for a suite of procedures defines their relationship to the calling program and to the order in which they are called. A detailed map of this information for the model program, CFAST, is given in Appendix C. It was felt that the structural details of the model portion of CFAST were most important since this is the program in the CFAST suite of programs that is most likely to be modified. The subroutine structure of the CFAST model is illustrated in Figure 11.

The model can be split into distinct parts. There are routines for reading data, calculating results and reporting the results to a file or printer. The major routines for performing these functions are identified in Figure 11. These physical interface routines link the CFAST model to the actual routines which calculate quantities such as mass or energy flow at one particular point in time for a given environment.

The routines SOLVE, RESID and DASSL are the key to understanding how the physical equations are solved. SOLVE is the control program that oversees the general solution of the problem. It invokes the differential equation solver DASSL [51] which calls RESID to solve the transport equations. The problem that these routines solve is as follows. Given a solution at time (t), what is the solution at time ($t + \Delta t$)? The transport equations are differential equations of the form

$$\begin{aligned} \frac{dy}{dt} &= f(y,t) \\ y(t_0) &= y_0 \end{aligned} \tag{89}$$

where y is a vector function representing pressure, layer height, mass, etc. and f is a vector function that represents changes in these values with respect to time. The term y_0 is an initial condition at the initial time t_0. The subroutine RESID (see sec. 4.3) computes the right hand side of eq (89) and returns a set of residuals of that calculation to be compared to the values expected by DASSL. DASSL then checks for convergence. Once DASSL reaches an error limit (defined as convergence of the equations) for the solution at ($t,t+\Delta t$), SOLVE then advances the solution of species concentration, wall temperature profiles, and mechanical ventilation for the same time interval.

Note that there are several distinct time scales that are involved in the solution of this type of problem. The fastest will be chemical kinetics. We avoid that scale by assuming that the chemistry is infinitely fast. The

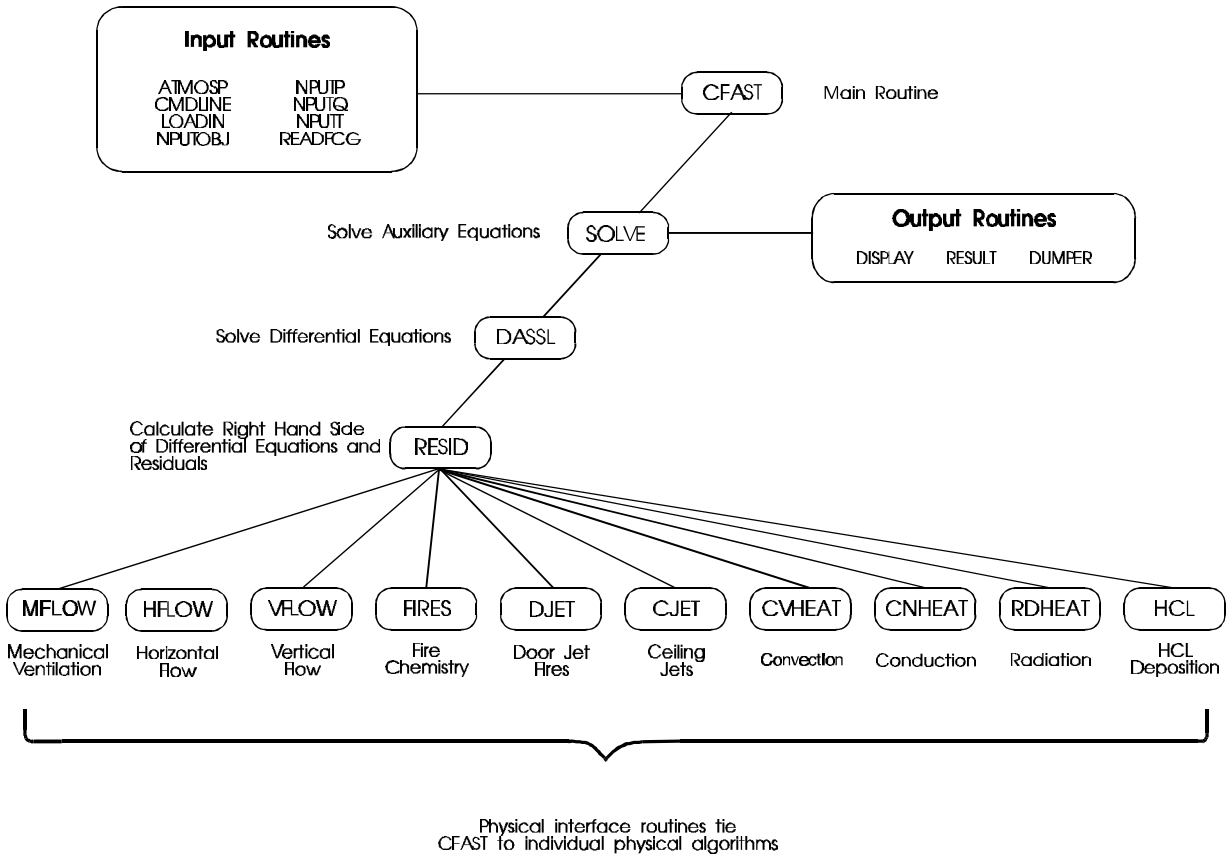


Figure 11. Subroutine Structure for the revised CFAST.

next larger time scale is that associated with the flow field. These are the equations which are cast into the form of ordinary differential equations. Then there is the time scale for mechanical ventilation, and finally, heat conduction through objects. By way of example, chemical kinetic times are typically on the order of milliseconds. The transport time scale will be on the order of 0.1 s. The mechanical ventilation and conduction time scales are typically several seconds, or even longer. Unlike earlier versions of the model, this new version dynamically adjusts the time step over the entire simulation to a value appropriate for the solution of the currently defined equation set. In addition to allowing a more correct solution to the pressure equation (without damping the rate of change of pressure), very large time steps are possible if the problem being solved approaches steady-state.

4.2 Data Flow

4.2.1 Input and Initialization

Command Line and Program Options: Each main module in the CFAST suite calls a set of routines which set up the physical environment for the model. The routines are READOP and OPENSHEL. They perform a number of housekeeping tasks. This information is related to the environment (computer platform) on which the model is running.

First, the command line is interpreted, providing file names and options. Up to two file names are available. The first entry is either an input file, or a configuration file. If it is an input file, then the name is provided in the variable "NNFILE" in the shell common block. It will not be opened. When a file is opened for input, the unit number IOFILI should be assigned. If there is a valid output file, its name is provided in the variable "OUTFILE." It will be opened, and the unit number is IOFILO. There is a default configuration file HV1.CF which will be read unless an alternative is provided on the command line. A configuration file name can be provided in place of the input file. In this case, the input file will be fetched from the variable DFILE from within the configuration file. In any case, the current configuration file will be named in CONFIG. Most modules will not run without the configuration file.

Second, the options are stored in the shell common blocks. There are currently four options available. All five are available to all modules. They are specified on the command line by

- (or /) option.

An example is the option to prevent the header from printing. This would be

-N.

The four which are presently read and decoded are

1. Report type (Rnn)
2. No header (N)
3. Turn on error logging (L)
4. Pass an environment file (Ffilename) - *used internally by CFAST to pass an environment file between modules*

These options are read and interpreted by the routine READOP.

Input Data File Processing: The data file is opened by the input routine, NPUTP. It is closed by this routine after all data has been read. There should be no units assigned, opened or closed while data retrieval is in process. Thus data entry should be done within the scope of NPUTP or NPUTQ only by these routines. Any data can be retrieved from the data files by NPUTQ. If subsidiary information is needed, then a reference file name should be read and stored in a variable kept in the PARAMS common block (or unlabeled common), and data fetches made after the *second* exit from NPUTP. This precludes initialization of such data during the geometry and fire specification process. The thermal properties are retrieved by the initialization routines at this point (after the restart return).

New key words that are to be added to the data file are placed in NPUTQ. It is important to follow the protocol as laid out therein, so that consistency checks can be performed on the data. Any physical initialization that needs to be done should be included in NPUTP, after the call to NPUTG. At this point, the model is completely set up. The only data that is not done are the total masses of the upper and lower layers. This is deferred to the originating routine. The name of the primary data file is in the variable NNFILE, but is open at this point, and pointing to the end of NNFILE. NPUTP can be referenced twice if a restart has been requested. In each case, NNFILE is closed prior to exiting.

In the process of inserting key words into NPUTQ, one will note that there are two case statements for the key words. The first is for a normal start, and the second is for a restart. In the latter case, some variables are

not, and should not, be valid. An example of the difference: the fire specification can change, but it makes no sense to change the physical layout, such as the number of compartments.

The next consideration is the setup provided for the physical system. Initialization is done by CFAST, or INITFS in the case of the data editor. Both preset memory, and call the routine NPUTP. NPUTP does the actual physical initialization. It in turn calls the routine NPUTQ which reads the data files. Subsequently, the geometry, species and graphics descriptors are set. Finally the environment is set by CFAST or INITFS. This includes reading the thermophysical properties and assigning them to the correct boundary, and all other auxiliary files as necessary. It is at this point that all the data files are closed, and the unit IOFILI is available. It should be closed, opened to the appropriate file, and *subsequently closed* if used for other purposes.

Problem Initialization: There are several logical switches that are set, based on the problem to be solved. The two most common are ACTIVS and SWITCH. The former is for active species. The latter serves two purposes, for active conduction and for miscellaneous parameters.

If a species is being computed, for whatever reason, then ACTIVS will be TRUE, otherwise it will be FALSE. This parameter is dimensioned to NS, the number of species which CFAST will follow. The order is

Table 4. Indices for Species Tracked by CFAST

INDEX	SPECIES	APPLICABLE KEY WORD
1	Nitrogen	none
2	Oxygen	O2
3	Carbon dioxide	none
4	Carbon monoxide	CO
5	Water vapor	none but HCR is related
6	Hydrogen chloride	HCL
7	Unburned hydrocarbons	none
8	Hydrogen cyanide	HCN
9	Soot	OD
10	Concentration time dose	CT (not a species)

For each species that is tracked, the variable ACTIVS(i) is set to true. There are two types of action that hinge on the setting of this variable. The first is in the availability and display of species information. The second is in the packing used in preparing the source terms for, and extracting them from the solver. The details of this activity are in the section on the **data copy** and RESID routines.

The variable SWITCH is used in two places. The first is to specify which boundaries in which compartments can conduct heat and where HCl deposition takes place. The parameter is set in NPUTQ, but verified by NPUTT. It can be set in NPUTQ if specified in the data file, but subsequently turned off by NPUTT if the name of the boundary can not be found in the thermophysical database. When the primary model is running, it will terminate if this latter condition is found, whereas the data editor will distinguish between the boundary being considered adiabatic with the name "OFF" and not found by "NONE." SWITCH is dimensioned NWAL by NR.

Since conduction is only allowed for NR-1 compartments, the last column can be used for miscellaneous variables. Once again, the default is false, but if the appropriate key word has been set, then the variable will be set to true.

(1,NR) - print the flow field and species - set in NPUTP; used only by the main model

The order of initialization is important. This is particularly true because of the caveat above that input/output units should not be assigned during the primary initialization. First, the main routine, CFAST or INITFS, initialize memory, and some physical constants such as the gravitational constant. The main data file is then opened. If one can not be found, then the model quits. The next step is to call NPUTP with the restart parameter of ISRSTR=1. NPUTP reads the header line to check for a correct file, then calls NPUTQ, with the ISRSTR=1. NPUTQ does all of the actual data entry, *via* unit IOFILI. After control is returned to NPUTP, physical initialization is done, for example setting the atmospheric ambient, calculating the volume of the compartments, and so forth. Then the graphics descriptors are read by LOADIN. These processes occur whether or not a restart will be done. Then control is returned to the main module. Some additional processing takes place to set the species of the ambient environment. If a restart has been requested, the appropriate history file is read for the requested interval. NPUTP rewinds the input file and once again calls NPUTQ, with ISRSTR=2. At this point there are some differences. Within NPUTQ, the case statement (discussed above) prevents some parameters from being reset.

Units are specific to the operating system. There are two general input/output units named IOFILI and IOFILO. In the current implementation they are numbered 1 and 6 respectively. There are additional units as follows:

1) ISRSTR tells us if this is a normal input file, or a restart

0 => Most data has been set - do initialization only
1 => Implies a normal read, with open
2 => Implies an update after restart

2) files used

1 => Configuration file, primary data file and data bases - opened and closed by each routine
2 => Help files
3 => Log file - open all the time
9 => History file i/o & font files - both are open intermittently
98 => ASCII history output

99 => "Other Objects" database

4.2.2 Data Structure and Data Flow Within CFAST

Information is passed between the various subroutines and main modules by the use of files and common blocks. The model has several common blocks associated with it. Of interest to most programmers is the way these data are used in the model. Appendices A through D provide details of the subroutines and variables in the CFAST model.

4.3 The Control Programs (SOLVE and RESID)

As discussed above, the routine RESID controls most of the model calculations. SOLVE coordinates the solution and output, but the physical phenomena are accessed by RESID. This section provides an annotated overview of these two control programs.

4.3.1 SOLVE

Since much of the function of the control routine SOLVE is bookkeeping, the source code is not particularly illustrative. Rather, for this routine, we will provide an summary of the functioning of the routine below.

1. Initialize the print, history, and plot times to the user's input specifications.

2. By calling routine INITSOLN, determine a set of initial pressures consistent with the initial conditions (temperatures and vent sizes) of the problem to be solved. By solving a set of linear equations to determine appropriate steady-state initial conditions for the pressures, the differential equation solver is able to determine solutions for the always difficult first second of the solution several times faster than allowing the differential equation solver to find the initial solution.

3. Output results (or initial conditions at time t=0) of the calculation by printing (routine RESULT), writing a history file interval (routine DUMPER), or plotting (routine DISPLAYC) results if current time is appropriate for such output.

4. Call the differential equation solver, DASSL, to advance the solution in time. The length of the advance in time is chosen dynamically by DASSL. DASSL chooses the time step but reports back a solution based on the lesser of the print, display and dump intervals. DASSL call RESID to compute the actual solution, as well as the residuals.

5. Advance the solution for species not handled directly by the differential equation solver by calling routines RESID (again, but with different switches) and TOXIC.

6. Repeat steps 3-5 until the final time is reached.

Note that many of the defaults, switches, and tolerances can be changed in the initialization routine. SOLVE calls the initialization routine INITSLV. It has switches built in which can be redefined with the configuration file, SOLVER.INI. If this file is not present, then internal defaults are used which are the best

available. However, for testing, turning off phenomena, alternative settings can be useful. See the appendices at the end for the format of this configuration file.

4.3.2 RESID

RESID is split into several parts. First, the current environment is copied from the form used by the differential equation solver into the environment common blocks for use by the physical routines in CFAST. Then the physical phenomena are calculated with calls to appropriate physical interface routines. Each physical interface routine returns its contribution to mass, enthalpy, and species flows into each layer in each room. These are then summed into total mass, enthalpy, and species flows into each layer in each room. Finally, the differential equations are formed for each room, wall surface, and mechanical ventilation system in the problem.

This portion of the model is the real numerical implementation and is accessed many times per simulation run. Careful thought must be given to the form of the routines since the execution time is *very* sensitive to the coding of the software. What follows is an annotated form of the routine RESID. Extraneous comments have been left out to shorten the listing somewhat.

```
      SUBROUTINE RESID(TSEC,X,XPSOLVE,DELTA,IRES,RPAR,IPAR)
```

Common blocks go here to define the environment for CFAST use by all physical routines. Definition of temporary variables used to store the output of each physical routine are also included here. See section 4.4 for the format of each of these variables. The routine DATACOPY is called to copy the environment from the form used by the differential equation solver into the environment common blocks for use by the physical routines in CFAST.

```
      XX0 = 0.0D0
      ND = 0
      NPROD = NLSPCT
      DT = TSEC - TOLD
C
      NIRM = NM1
C
      CALL DATACOPY(X,ODEVARA+ODEVARB)
```

The IPAR and RPAR parameters are passed from SOLVE to RESID via DASSL and are used to control the calculation of the residuals by RESID. For a call to RESID from the differential equation solver, DASSL, IPAR(2) is equal to the parameter SOME to indicate that the routine is to calculate the set of differential equations without including the species. Species are updated by SOLVE once DASSL has found an appropriate solution for the smaller equation set.

```
      IF (IPAR(2).EQ.SOME) THEN
        UPDATE = .FALSE.
      ELSE
        UPDATE = .TRUE.
      END IF
      EPSP = RPAR(1)
```

All of the physical phenomena included in the model are included here with calls to the physical interface routines for each phenomena. Each physical interface routine returns its contribution to mass, enthalpy, and species flows into each layer in each room.

```
C
C     CALCULATE FLOW THROUGH VENTS (HFLOW FOR HORIZONTAL FLOW
C     THROUGH VERTICAL VENTS, VFLOW FOR VERTICAL FLOW THROUGH
C     HORIZONTAL VENTS, AND MVENT FOR MECHANICAL VENTILATION)
C
      CALL HFLOW(TSEC,EPSP,NPROD,FLWNVNT,QLPQUV)
      CALL VFLOW(FLWHVNT,QLPQUH)
      CALL MVENT(X(NOFPMV+1),X(NOFTMV+1),XPSOLVE(NOFTMV+1),FLWMV,
     +    DELTA(NOFPMV+1),DELTA(NOFTMV+1),XPRIME(NOFHVPR+1),NPROD)
C
C     CALCULATE HEAT AND MASS FLOWS DUE TO FIRES
C
      CALL FIRES(TSEC,FLWF,QLPQUF,NFIRE,IFROOM,XFIRE)
      CALL SORTFR(NFIRE,IFROOM,XFIRE,IFRPNT)
      CALL DJET(NFIRE,FLWDJF,XFIRE)
C
C     CALCULATE FLOW AND FLUX DUE TO HEAT TRANSFER (CEILING JETS,
C     CONVECTION AND RADIATION
C
      CALL CJET(IFRPNT,XFIRE,ND,XD,YD,ZD,FLWCJET,FLXCJET,TD,VD)
      CALL CVHEAT(IFRPNT,FLWCV,FLXCV)
      CALL RDHEAT(IFRPNT,XFIRE,FLWRAD,FLXRAD)
C
C     CALCULATE HCL DEPOSITION TO WALLS
C
      CALL HCL(FLWHCL,FLXHCL)
```

The flows returned from each physical interface routine are then summed into total mass, enthalpy, and species flows into each layer in each room. The form of each of these flows is discussed in section 4.4. In general, the array FLWTOT(room,species,layer) contains the total flow of each species into each layer of each room in the simulation. For ease of definition, mass and enthalpy are included in this array as pseudo-species (1 & 2) and summed along with the actual species (3 to 2+lsp). Heat flux to surfaces is included in a similar manner for used by the conduction routine.

```
C
C     SUM FLOW FOR INSIDE ROOMS
C
      DO 50 IROOM = 1, NIRM
        QLPQUR(IROOM) = QLPQUV(IROOM) + QLPQUH(IROOM) + QLPQUF(IROOM) +
     +      FLWCV(IROOM,LL) + FLWCV(IROOM,UU) + FLWRAD(IROOM,LL) +
     +      FLWRAD(IROOM,UU) + FLWCJET(IROOM,LL) + FLWCJET(IROOM,UU) +
     +      FLWDJF(IROOM,Q,LL) + FLWDJF(IROOM,Q,UU) +
     +      FLWMV(IROOM,Q,LL) + FLWMV(IROOM,Q,UU)
        DO 40 IPROD = 1, NPROD + 2
          IP = IZPMAP(IPROD)
          FLWTOT(IROOM,IPROD,LL) = FLWNVNT(IROOM,IPROD,LL) +
     +        FLWMV(IROOM,IP,LL) + FLWF(IROOM,IP,LL) +
     +        FLWDJF(IROOM,IP,LL) + FLWHVNT(IROOM,IP,LL)
          FLWTOT(IROOM,IPROD,UU) = FLWNVNT(IROOM,IPROD,UU) +
     +        FLWMV(IROOM,IP,UU) + FLWF(IROOM,IP,UU) +
     +        FLWDJF(IROOM,IP,UU) + FLWHVNT(IROOM,IP,UU)
   40   CONTINUE
C
C     ADD IN HCL CONTRIBUTION TO FLWTOT
C
        IF (ACTIVS(6)) THEN
          FLWTOT(IROOM,1,LL) = FLWTOT(IROOM,1,LL) + FLWHCL(IROOM,1,LL)
          FLWTOT(IROOM,1,UU) = FLWTOT(IROOM,1,UU) + FLWHCL(IROOM,1,UU)
          FLWTOT(IROOM,8,LL) = FLWTOT(IROOM,8,LL) + FLWHCL(IROOM,8,LL)
          FLWTOT(IROOM,8,UU) = FLWTOT(IROOM,8,UU) + FLWHCL(IROOM,8,UU)
        END IF
C
        FLWTOT(IROOM,Q,LL) = FLWTOT(IROOM,Q,LL) + FLWCV(IROOM,LL) +
     +      FLWRAD(IROOM,LL) + FLWCJET(IROOM,LL)
        FLWTOT(IROOM,Q,UU) = FLWTOT(IROOM,Q,UU) + FLWCV(IROOM,UU) +
     +      FLWRAD(IROOM,UU) + FLWCJET(IROOM,UU)
C
   50 CONTINUE
C
C     SUM FLUX FOR INSIDE ROOMS
```

```
C
      DO 70 IROOM = 1, NIRM
        DO 60 IWALL = 1, NWAL
          IF (SWITCH(IWALL,IROOM)) THEN
            FLXTOT(IROOM,IWALL) = FLXCV(IROOM,IWALL) +
     +          FLXRAD(IROOM,IWALL) + FLXCJET(IROOM,IWALL)
          END IF
   60   CONTINUE
   70 CONTINUE
```

The differential equations are formed for each room, wall surface, and mechanical ventilation system in the problem. These follow directly from the derivation in section 2.2.

```
      DO 80 IROOM = 1, NIRM
        AROOM = AR(IROOM)
        HCEIL = HR(IROOM)
        PABS = ZZPABS(IROOM)
        HINTER = ZZHLAY(IROOM,LL)
        QL = FLWTOT(IROOM,Q,LL)
        QU = FLWTOT(IROOM,Q,UU)
        TMU = FLWTOT(IROOM,M,UU)
        TML = FLWTOT(IROOM,M,LL)
        QLPQU = QLPQUR(IROOM)
C
C     PRESSURE EQUATION
C
        PDOT = (GAMMA-1.0D0) * QLPQU / (AROOM*HCEIL)
        XPRIME(IROOM) = PDOT
C
C     UPPER LAYER TEMPERATURE EQUATION
C
        TLAYDU = (QU-CP*TMU*ZZTEMP(IROOM,UU)) / (CP*ZZMASS(IROOM,UU))
        IF (OPTION(FODE).EQ.ON) THEN
          TLAYDU = TLAYDU + PDOT / (CP*ZZRHO(IROOM,UU))
        END IF
        XPRIME(IROOM+NOFTU) = TLAYDU
C
C     UPPER LAYER VOLUME EQUATION
C
        VLAYD = (GAMMA-1.0D0) * QU / (GAMMA*PABS)
        IF (OPTION(FODE).EQ.ON) THEN
          VLAYD = VLAYD - ZZVOL(IROOM,UU) * PDOT / (GAMMA*PABS)
        END IF
        XPRIME(IROOM+NOFVU) = VLAYD
C
C     LOWER LAYER TEMPERATURE EQUATION
C
        TLAYDL = (QL-CP*TML*ZZTEMP(IROOM,LL)) / (CP*ZZMASS(IROOM,LL))
        IF (OPTION(FODE).EQ.ON) THEN
          TLAYDL = TLAYDL + PDOT / (CP*ZZRHO(IROOM,LL))
        END IF
        XPRIME(IROOM+NOFTL) = TLAYDL
   80 CONTINUE
```

The species are only calculated once DASSL has an acceptable solution for the equation set not including the species. We presume that the species production rates occur on a time scale similar to the total mass production which is solved directly.

```
      IF (NPROD.GT.0.AND.IPAR(2).EQ.ALL) THEN
        IPRODU = NOFPRD - 1
        DO 100 IPROD = 1, NPROD
          DO 90 IROOM = 1, NM1
            HCEIL = HR(IROOM)
            HINTER = ZZHLAY(IROOM,LL)
            IPRODU = IPRODU + 2
            IPRODL = IPRODU + 1
            IF (HINTER.LT.HCEIL) THEN
              XPRIME(IPRODU) = FLWTOT(IROOM,IPROD+2,UU)
            ELSE IF (HINTER.GE.HCEIL.AND.FLWTOT(IROOM,IP,UU).LT.XX0)
     +          THEN
```

```
            XPRIME(IPRODU) = FLWTOT(IROOM,IPROD+2,UU)
          ELSE
            XPRIME(IPRODU) = XX0
          END IF
          IF (HINTER.GT.XX0) THEN
            XPRIME(IPRODL) = FLWTOT(IROOM,IPROD+2,LL)
          ELSE IF (HINTER.LE.XX0.AND.FLWTOT(IROOM,IP,LL).GT.XX0) THEN
            XPRIME(IPRODL) = FLWTOT(IROOM,IPROD+2,LL)
          ELSE
            XPRIME(IPRODL) = XX0
          END IF
   90     CONTINUE
  100   CONTINUE
      END IF
```

Finally, the residuals are calculated. These are simply the difference between the solution vector calculated by RESID and that passed to RESID by the differential equation solver. This is done in several parts to correspond with the layout of the solution vector.

```
C
C     RESIDUALS FOR PRESSURE, LAYER VOLUME, AND LAYER TEMPERATURES
C
      DO 110 I = NOFP + 1, NOFP + NM1
        DELTA(I) = XPRIME(I) - XPSOLVE(I)
  110 CONTINUE
      DO 120 I = NOFTU + 1, NOFTU + 3 * NM1
        DELTA(I) = XPRIME(I) - XPSOLVE(I)
  120 CONTINUE
C
C     CONDUCTION RESIDUAL
C
      CALL CNHEAT(UPDATE,DT,FLXTOT,DELTA)
C
C     RESIDUALS FOR GAS LAYER SPECIES, NOTE THAT DASSL IS NOT SOLVING FOR
C     SPECIES NOW, THIS IS DONE IN SOLVE
C
      DO 130 I = NOFPRD + 1, NOFPRD + 2 * NPROD * NM1
        DELTA(I) = XPRIME(I) - XPSOLVE(I)
  130 CONTINUE
C
C     RESIDUAL FOR HVAC SPECIES
C
      IF (NPROD.NE.0) THEN
        DO 140 I = NOFHVPR + 1, NOFHVPR + NLSPCT * NHVSYS
          DELTA(I) = XPRIME(I) - XPSOLVE(I)
  140   CONTINUE
      END IF
      IF (IPAR(2).EQ.SOME) THEN
        NPROD = NPRODSV
      END IF
      RETURN
      END
```

4.4 Interface to the CFAST Physical Interface Routines

Each physical interface routine calculates flow and/or flux terms as appropriate for all rooms and/or surfaces of the simulation being modeled. These flow and flux terms are the effect of the phenomenon on each of the layers and/or surfaces and includes flows due to mass, enthalpy and products of combustion. Rather than using multiple variables for each room, these are organized into a single array for each phenomenon. This structure is shown in Figure 12. To illustrate the organization of the physical interface routines, the following outlines the steps in calculating one of the phenomena.

The physical interface routine, FIRES, calculates the rates of addition of mass, enthalpy, and species into all layers in all rooms from all fires in a simulation. For each fire, the following scheme is employed:

1. Initialize the fire data structure, FLWF, to zero.

2. For each specified fire, the routine PYROLS (for the main fire) or OBJINT (for object fires) calculates time dependent quantities for the time of interest by interpolating between the time points specified by the user. The routine DOFIRE calculates the plume entrainment rate.

3. For a type 1 (unconstrained) fire, the routine DOFIRE sets the burning rate to the pyrolysis rate. The heat release rate is found by multiplying the burning rate by the heat of combustion.

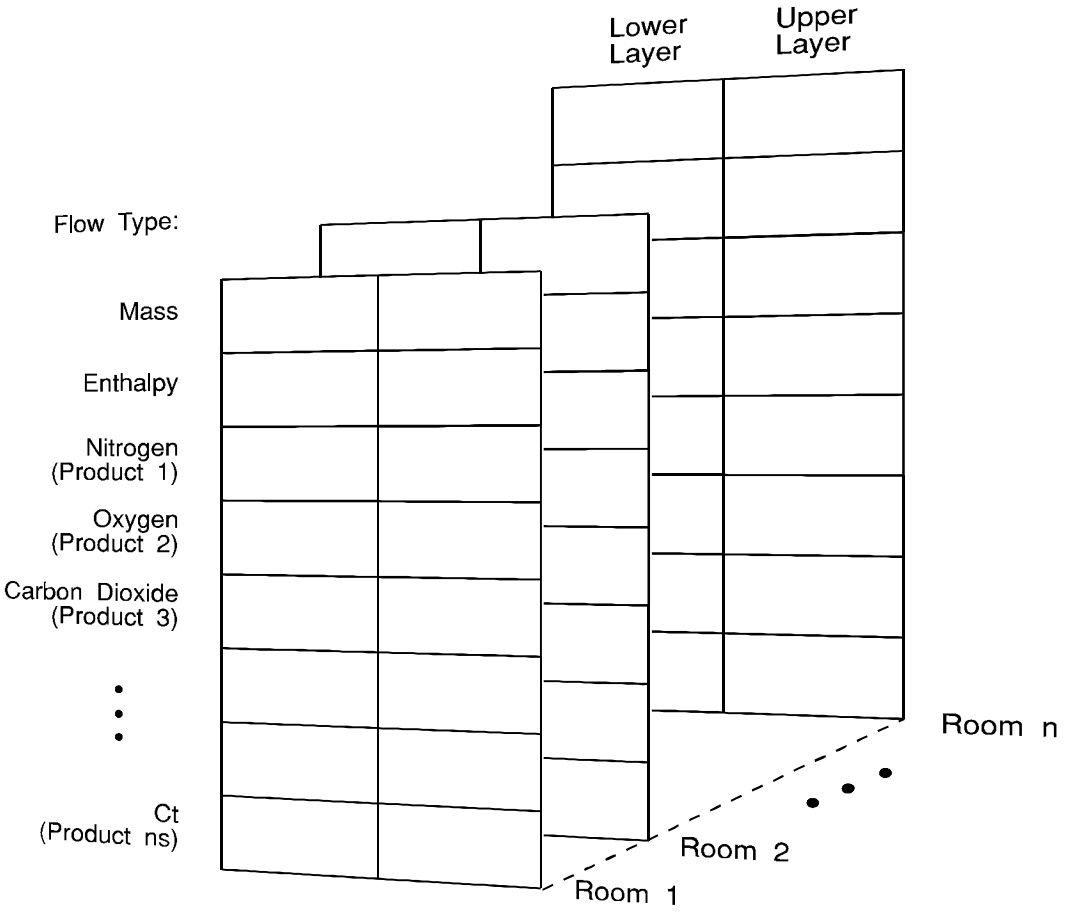


Figure 12. Data structure for flow and/or flux terms returned from physical interface routines to the control routine RESID.

4. For a type 2 (constrained) fire, the prescribed chemistry scheme, discussed above, is used to constrain the burning rate based on *both* the fuel and oxygen available. This chemistry scheme is implemented in the routine CHEMIE. This calculation is done for both the lower layer (from the mass entrained by the plume) and for burning in the upper layer (with oxygen and fuel available in the layer).

5. Sum the contributions from all fires into the fire data structure for return to the control routine. The following code fragment is typical of those in all of the physical interface routines:

```
      FLWF(LFBO,M,UPPER) = FLWF(LFBO,M,UPPER) + EMS(LFBO)
      FLWF(LFBO,M,LOWER) = FLWF(LFBO,M,LOWER) - EME(LFBO)
      FLWF(LFBO,Q,UPPER) = FLWF(LFBO,Q,UPPER) + QF(LFBO) + QEME + QEMP
      FLWF(LFBO,Q,LOWER) = FLWF(LFBO,Q,LOWER)            - QEME
      QLPQUF(LFBO)       = QLPQUF(LFBO)       + QF(LFBO)        + QEMP
      DO 40 LSP = 1, NS
        FLWF(LFBO,LSP+2,UPPER) = FLWF(LFBO,LSP+2,UPPER) +
     +       XNTMS(UPPER,LSP)
        FLWF(LFBO,LSP+2,LOWER) = FLWF(LFBO,LSP+2,LOWER) +
     +       XNTMS(LOWER,LSP)
40    CONTINUE
```

5 Verification of the Model

5.1 Available Experimental Data

Several systematic test series have been undertaken specifically to provide data for comparison with model predictions. In other cases, tests in which fire properties have been systematically varied (for various reasons) have been modeled using current computer fire simulations. In the first group are the study of Alpert et al. [52] for a single room connected to a short, open corridor, and those of Cooper et al. [53] and Peacock et al. [18] for gas burner fires in a room-corridor-room configuration. Although the second group is large, the works of Quintiere and McCaffrey [54], and Heskestad and Hill [55] are particularly detailed.

Cooper et al. [53] reported an experimental study of the dynamics of smoke filling in realistic, full-scale, multi-room fire scenarios. A major goal of the study was to generate an experimental database for use in the verification of mathematical fire simulation models. The test space involved 2 or 3 rooms, connected by open doorways. During the study, the areas were partitioned to yield four different configurations. One of the rooms was a burn room containing a methane burner which produced either a constant heat release rate of 25, 100, or 225 kW or a time-varying heat release rate which increased linearly with time from zero at ignition to 300 kW in 600 s. An artificial smoke source near the ceiling of the burn room provided a means for visualizing the descent of the hot layer and the dynamics of the smoke filling process in the various spaces. The development of the hot stratified layers in the various spaces was monitored by vertical arrays of thermocouples and photometers. A layer interface was identified and its position as a function of time was determined. An analysis and discussion of the results including layer interface position, temperature, and doorway pressure differentials is presented. These data were later used by Rockett et al. [56], [57] for comparison to a modern predictive fire model [58].

Quintiere and McCaffrey [54] described a series of experiments designed to provide a measure of the behavior of cellular plastics in burning conditions related to real life. They experimentally determined the effects of fire size, fuel type, and natural ventilation conditions on the resulting room fire variables, such as temperature, radiant heat flux to room surfaces, burning rate, and air flow rate. This was accomplished by burning up to four cribs made of sugar pine or of a rigid polyurethane foam to provide a range of fire sizes intended to simulate fires representative of small furnishings to chairs of moderate size. Although few replicates were included in the test series, fuel type and quantity, and the room door opening width were varied. The data from these experiments were analyzed with quantities averaged over the peak burning period to yield the conditions for flashover in terms of fuel type, fuel amount, and doorway width. The data collected were to serve as a basis for assessing the accuracy of a mathematical model of fire growth from burning cribs.

Heskestad and Hill [55] performed a series of 60 fire tests in a room/corridor configuration to establish accuracy assessment data for theoretical fire models of multi-room fire situations with particular emphasis on health care facilities. With steady state and growing fires from 56 kW to 2 MW, measurements of gas temperatures, ceiling temperatures, smoke optical densities, concentrations of CO, CO_2, and O_2, gas velocities, and pressure differentials were made. Various combinations of fire size, door opening size, window opening size, and ventilation were studied. To increase the number of combinations, only a few replicates of several of the individual test configurations were performed.

Except for the data of Cooper et al. [53] and Quintiere and McCaffrey [54] which are not available in machine readable form, the above data, along with other experimental results, have been reviewed by

Peacock, Davis and Babrauskas [59]. They provide a single consistent form for the experimental data from several series of experiments. Five sets of experimental data which can be used to test the limits of a typical two-zone fire model are detailed. Availability of ancillary data (such as smaller-scale test results) is included. These descriptions, along with the data should allow comparisons between the experiment and model predictions. The base of experimental data ranges in complexity from one-room tests with individual furniture items to a series of tests conducted in a multiple-story hotel equipped with a zoned smoke control system. These data will be used as the set of experimental results for comparisons in this paper.

5.2 Previous Comparisons with Experimental Data

Several researchers have studied the level of agreement between computer fire models and real-scale fires. These comparisons fall into two broad categories: fire reconstruction and comparison with laboratory experiments. Both categories provide a level of verification for the models used. Fire reconstruction, although often more qualitative, provides a higher degree of confidence for the user when the models successfully simulate real-life conditions. Comparisons with laboratory experiments, however, can yield detailed comparisons that can point out weaknesses in the individual phenomena included in the models.

Fire reconstructions: Nelson [60] used simple computer fire models along with existing experimental data to develop an analysis of a large high-rise building fire. This analysis showed the value of available analytical calculations in reconstructing the events involved in a multiple-story fire. Bukowski [61] has applied the FAST model (an earlier version of the CFAST model) in a litigation against the United States Government. At the request of the Justice Department, the model was used to recreate a multiple-fatality fire in a residence. The analysis reproduced many details of the fire including conditions consistent with damage patterns to the building, the successful escape of three older children, and three fatalities including the locations of the bodies and the autopsy results. Emmons applied computer fire modeling to the MGM Grand Hotel fire of 1980. This work, conducted during the litigation of this fire was only recently published [62]. Using the HARVARD 5 model, Prof. Emmons analyzed the relative contributions of booth seating, ceiling tiles, decorative beams, and the HVAC system on the outcome of the fire.

Comparisons with laboratory experiments: Rockett, Morita, and Cooper [57] used the HARVARD VI multi-room fire model to simulate the results of real-scale, multi-room fire experiments. These experiments can be characterized by fire sizes of several hundred kW and total compartment volume of about 1000 m^3. While the model was generally found to provide favorable simulations, several areas where improvements were needed were identified. They pointed out limitations in modeling of oxygen-limited burning, mixing of gases at vents, convective heat transfer, and plume entrainment.

Jones and Peacock [9] presented a limited set of comparisons between the FAST model and a multi-room fire test. The experiment involved a constant fire of about 100 kW in a three-compartment configuration of about 100 m^3. They noted “slight over-prediction” of the upper layer temperature and satisfactory prediction of the layer interface position. Again, convective heating and plume entrainment were seen to limit the accuracy of the predictions. A comparison of predicted and measured pressures in the rooms showed good agreement. Since pressure is the driving force for flow between compartments, this agreement was seen as important.

Levine and Nelson [63] used a combination of full-scale fire testing and modeling to simulate a fire in a residence. The 1987 fire in a first-floor kitchen resulted in the deaths of three persons in an upstairs bedroom, one with a reported blood carboxyhemoglobin content of 91 percent. Considerable physical

evidence remained. The fire was successfully simulated at full scale in a fully-instrumented seven-room two-story test structure. The data collected during the test have been used to test the predictive abilities of two multiroom computer fire models: FAST and HARVARD VI. A coherent ceiling layer flow occurred during the full-scale test and quickly carried high concentrations of carbon monoxide to remote compartments. Such flow is not directly accounted for in either computer code. However, both codes predicted the carbon monoxide buildup in the room most remote from the fire. Prediction of the pre-flashover temperature rise was also good. Prediction of temperatures after flashover that occurred in the room of fire origin was less good. Other predictions of conditions throughout the seven test rooms varied from good approximations to significant deviations from test data. Some of these deviations are believed to be due to phenomena not considered in any computer models.

Deal [64] reviewed four computer fire models (CCFM [10], FIRST [6], FPETOOL [65] and FAST) to ascertain the relative performance of the models in simulating fire experiments in a small room (about 12 m^3 in volume) in which the vent and fuel effects were varied. Peak fire size in the experiments ranged up to 800 kW. All the models simulated the experimental conditions including temperature, species generation, and vent flows, quite satisfactorily. With a variety of conditions, including narrow and normal vent widths, plastic and wood fuels, and flashover and sub-flashover fire temperatures, competence of the models at these room geometries was demonstrated.

Duong [66] studied the predictions of several computer fire models (CCFM, FAST, FIRST, and BRI [7]), comparing the models with one another and with large fires (4 to 36 MW) in an aircraft hanger (60,000 m^3). For the 4 MW fire size, he concluded that all the models are reasonably accurate. At 36 MW, however, none of the models did well. Limitations of the heat conduction and plume entrainment algorithms were seen to account for some of the inaccuracies.

5.3 Model Parameters Selected for Comparison

Comparisons of model predictions with experimental measurements serves two purposes: 1) to determine, within limits, the accuracy of the predictions for those quantities of interest to the users of the models (usually those extensive variables related to hazard), and 2) to highlight the strengths and weaknesses of the underlying algorithms in the models to guide future improvements in the models. The predicted variables selected for comparison must deal with both of these purposes.

Most of the studies discussed above present a consistent set of variables of interest to the model user: gas temperature, gas species concentrations, and layer interface position. To assess the accuracy of the physical basis of the models, additional variables must be included. Pressure drives the movement of gases through openings. The pyrolysis rate, and heat release rate of the fire in turn, produces the gases of interest to be moved.

In this section, we will consider all these variables for comparison:

- upper and lower layer gas temperature,
- layer interface position,
- gas species concentration.
- fire pyrolysis and heat release rate,
- room pressure, and
- vent flow,

Although there are certainly other comparisons of interest, these will provide an indication of the match of the model to the experimental data.

5.4 Experimental Data Selected for Comparison

A total of five different real-scale fire tests were selected for the current comparisons to represent a range of challenges for the CFAST model:

1) A single-room test using upholstered furniture as the burning item was selected for its well-characterized and realistic fire source in a simple single-room geometry [67]. Figure 13 shows the room and instrumentation used during the tests. Heat release rate, mass loss rate, and species yields are available for the test. This should allow straightforward application of the model. Peak fire size was about 2.9 MW with a total room volume of 21 m^3.

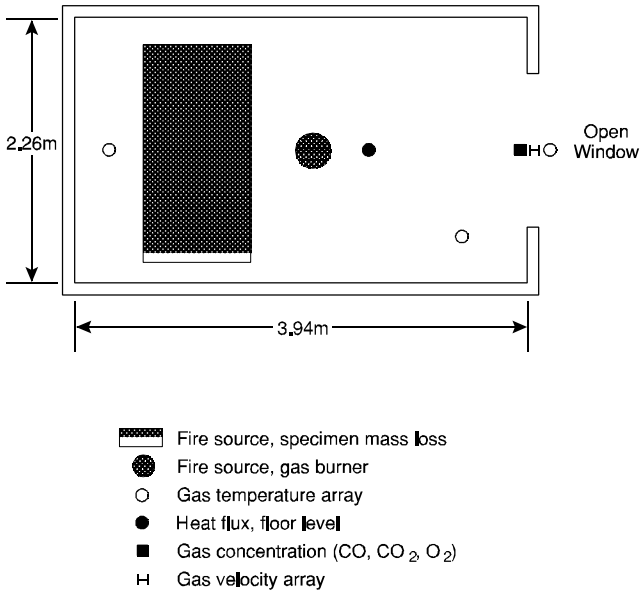


Figure 13. Single-room furniture fire test.

2) Like the first test, this test is a single-room fire test using furniture as the fire source [68]. It expands upon that data set by adding the phenomenon of wall burning. Figure 14 shows the test room and exhaust hood arrangement. Peak fire size was about 7 MW. Room size is similar to the first test.

3) This data set is actually an average of a series of 11 replicate tests in a three-room configuration with simple steady-state gas burner fires [18]. Figure 15 shows the room configuration and instrumentation for the test. It provides a basic set of quantities that are predicted by current fire models for small to medium size fires. Since all fires were gas burner fires, simulation should be straightforward. It is of particular interest since it was undertaken as a part of a program to develop a methodology for the evaluation and accuracy assessment of fire models. Fire size was about 100 kW with a total volume of 100 m^3.

4) This data set is part a series of tests conducted in a multiple room configuration with more complex gas burner fires than the previous data set [55] , [69]. This study was included because it expands upon that data set by providing larger and time-varying gas burner fires in a room-corridor configuration. Figure 16 shows the room configuration and instrumentation for the test. Fire size was about up to 1 MW with a total volume of 200 m^3.

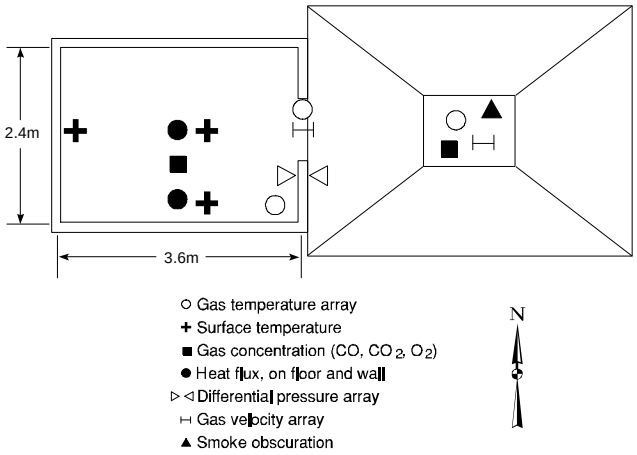


Figure 14. Single-room furniture fire test with wall burning.

5) By far the most complex test, this data set is part of a series of full-scale experiments conducted to evaluate zoned smoke control systems, with and without stairwell pressurization [70]. It was conducted in a seven story hotel with multiple rooms on each floor and a stairwell connecting to all floors. Figure 17 shows the room configuration and instrumentation for one of the eight floors of the building. This data set was chosen because it would be considered beyond the scope of most current fire models. Measured temperatures and pressure differences between the rooms and floors of the building are extensive and consistent. Peak fire size was 3 MW with a total building volume of 140,000 m^3.

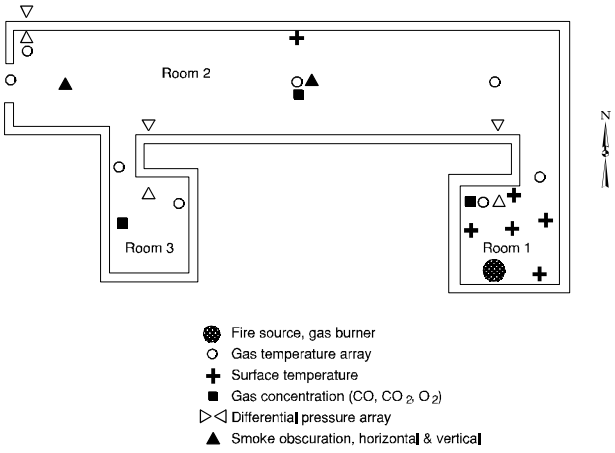


Figure 15. Three-room gas burner tests with a corridor.

5.5 Discussion

All of the simulations were performed with the CFAST model on an MS-DOS** compatible computer. For each of the data sets, the model data were developed from the building and fire descriptions provided in the original reports. Obtaining building geometry, construction materials, and room interconnections was straightforward. Usually, description of the fire source was more difficult. Where freeburn data were available, such data were used to describe the heat release rate, pyrolysis rate, and species yields. In other cases, estimates from tests of similar materials or textbook values were used to determine missing quantities.

** The use of company names or trade names within this report is made only for the purpose of identifying those computer hardware or software products with which the compatibility of the CFAST programs has been tested. Such use does not constitute any endorsement of those products by the National Institute of Standards and Technology.

How to best quantify the comparisons between model predictions and experiments is not obvious. The necessary and perceived level of agreement for any variable is dependent upon both the typical use of the variable in a given simulation (for instance, the user may be interested in the time it takes to reach a certain temperature in the room), the nature of the experiment (peak temperatures would be of little interest in an experiment which quickly reached steady state), and the context of the comparison in relation to other comparisons being made (a true validation of a model would involve proper statistical treatment of many compared variables).

Insufficient experimental data and understanding of how to compare the numerous variables in a complex fire model prevent a true validation of the model. Thus, the comparisons of the differences between model predictions and experimental data in this paper are intentionally simple and vary from test to test and from variable to variable due to the changing nature of the tests and typical use of different variables.

5.5.1 Layer Temperature and Interface Position

Arguably the most frequent question asked about a fire is "How hot did it become?" Temperature in the rooms of a structure is an obvious indicator to answer this question. Peak temperature, time to peak temperature, or time to reach a chosen temperature tenability limit are typical values of interest. Quality of the prediction (or measurement) of layer interface position is more difficult to quantify. Although observed valid in a range of experiments, the two-layer assumption is in many ways just a convenience for modeling. From a standpoint of hazard, time of descent to a chosen level may be a reasonable criterion (assuming some in the room will then either be forced to crawl beneath the interface to breathe the "clean" atmosphere near the floor or be forced to breath the upper layer gases). Minimum values may also be used

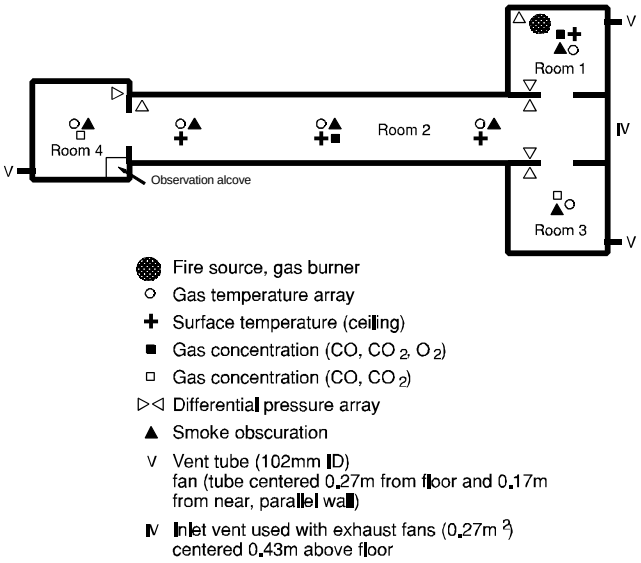


Figure 16. Four-room gas burner tests with a corridor.

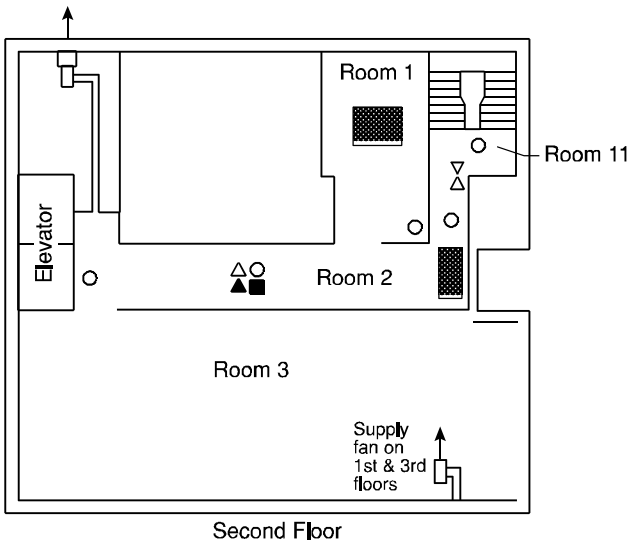


Figure 17. Second floor for multiple-story building tests.

to indicate general agreement. For the single-room tests with furniture or wall-burning, these are appropriate indicators to judge the comparisons between model and experiment. For the more-closely steady-state three- and four-room tests with corridor or the multiple-story building tests, a steady-state average better characterizes the nature of the experiment.

Figures 18-20 and Tables 5-7 show the upper layer temperature, lower layer temperature, and interface position for the tests studied. Like all zone-based fire models, CFAST calculates conditions within each room as an upper and a lower volume (layer), each with uniform conditions throughout the volume at any instant of time. Thus, for the model, the temperature environment within a room can be described by an upper and lower layer temperature and by the position of the interface between these two layers. By contrast, experimental measurements often take the form of a vertical array of measurement points describing a profile of temperature. Techniques for collapsing these profiles to data that can be compared to zone fire models are available [59] and are used here to facilitate the comparison. (See Figure 21 for an example of the calculated and experimentally measured temperature profiles.)

For the single-room tests, predicted temperatures and layer interface position show obvious similarities to the measured values. Peak values occur at similar times with comparable rise and fall for most comparisons. Interface height for the single-room with wall-burning is a notable exception. Unlike the model prediction, the experimental measurement does not show the rise and fall in concert with the temperature measurement. Peak values are typically higher for upper layer temperature and lower for lower layer temperature and layer interface position. For all the tests, including the single-room tests, times to peak values and times to 100 °C predicted by the model average within 25 s of experimentally measured values.

Numbers in parentheses are model predictions	Peak Value (°C)	Time to Peak (s)	Time to 100°C (s)	Steady-State Value (°C)	Similar Shape?
Single-room furniture tests[a] (Tests 1 and 6)	790 (970) 920 (970)	500 (510) 450 (510)	290 (310) 290 (310)	--[b]	✓
	590 (790) 900 (790)	510 (510) 510 (510)	330 (340) 330 (340)	--	✓
Single-room tests with wall burning (Tests 1 and 2)	750 (710)	710 (700)	100 (120)	--	✓
	810 (1550)	520 (470)	100 (70)	--	✓
Three-room tests with corridor[c] (SET 4, 11 replicates)	--	--	100 (110) 830 (n.r.) n.r.	230 (250) 75 (90) 45 (45)	
Four-room tests with corridor[c] (Tests 19 and 21)	--	--	195 (190) n.r. (270) n.r. n.r.	240 (470) 70 (110) 55 (35) 40 (35)	✓
	--	--	200 (190) n.r. (230) n.r. n.r.	260 (440) 80 (140) 65 (60) 50 (60)	
Multiple-story building (Test 7)	--	--	390 (375) 210 (150) n.r.	270 (340) 110 (65) 15 (15)	✓

[a] Two measurement positions within the room were available from the experimental data.
[b] Not appropriate for the experiment.
[c] Multiple entries indicate multiple comparable rooms in the test structure.

Table 5. Comparison of experimental measurements and model predictions of upper layer temperature (°C) for several tests

Numbers in parentheses are model predictions	Peak Value (°C)	Time to Peak (s)	Time to 100°C (s)	Steady-State Value (°C)	Similar Shape?
Single-room furniture tests[a]	570 (650) 590 (650)	500 (510) 420 (510)	370 (380) 390 (380)	--[b]	✓
	230 (340) 590 (340)	510 (510) 500 (510)	410 (440) 390 (440)	--	✓
Single-room tests with wall burning	710 (250)	710 (700)	240 (220)	--	✓
	700 (620)	520 (450)	290 (290)	--	✓
Three-room tests with corridor[c]	--	--	n.r. n.r. n.r.	70 (40) 30 (30) 23 (30)	
Four-room tests with corridor[c]	--	--	n.r. n.r. n.r.	75 (45) 21 (19) 21 (15)	✓
	--	--	n.r. n.r. n.r.	70 (32) 20 (17) 20 (15)	
Multiple-story building	--	--	400 (n.r.) n.r. n.r.	40 (37) 85 (70) 14 (16)	✓

See notes for Table 1.

Table 6. Comparison of experimental measurements and model predictions of lower layer temperature (°C) for several tests

Numbers in parentheses are model predictions	Peak Value (m)	Time to Peak (s)	Time to 1 m (s)	Steady-State Value (m)	Similar Shape?
Single-room furniture tests[a]	0.8 (0.7) 0.8 (0.7)	420 (480) 450 (480)	400 (400) (380) 400	--[b]	✓
	0.8 (0.6) 0.9 (0.6)	480 (510) 460 (510)	420 (430) 430 (430)	--	✓
Single-room tests with wall burning	0.2 (0.7)	710 (230)	120 (210)	--	✓
	0.1 (0.6)	500 (410)	80 (280)	--	✓
Three-room tests with corridor[c]	--	--	360 (n.r.) 1210 (n.r.) 90 (270)	1.0 (1.5) 1.2 (1.3) 0.9 (0.7)	✓
Four-room tests with corridor[c]	--	--	n.a.	0.7 (1.7) 1.0 (1.6) 1.0 (1.7) 0.7 (1.7)	
	--	--	n.a.	0.8 (1.1) 0.9 (1.1) 0.8 (1.0) 0.6 (1.0)	
Multiple-story building	--	--	n.a.	0.3 (1.8) 0.8 (2.1) 1.8 (1.8)	

See notes for Table 1.

Table 7. Comparison of experimental measurements and model predictions of layer interface position (m) for several tests

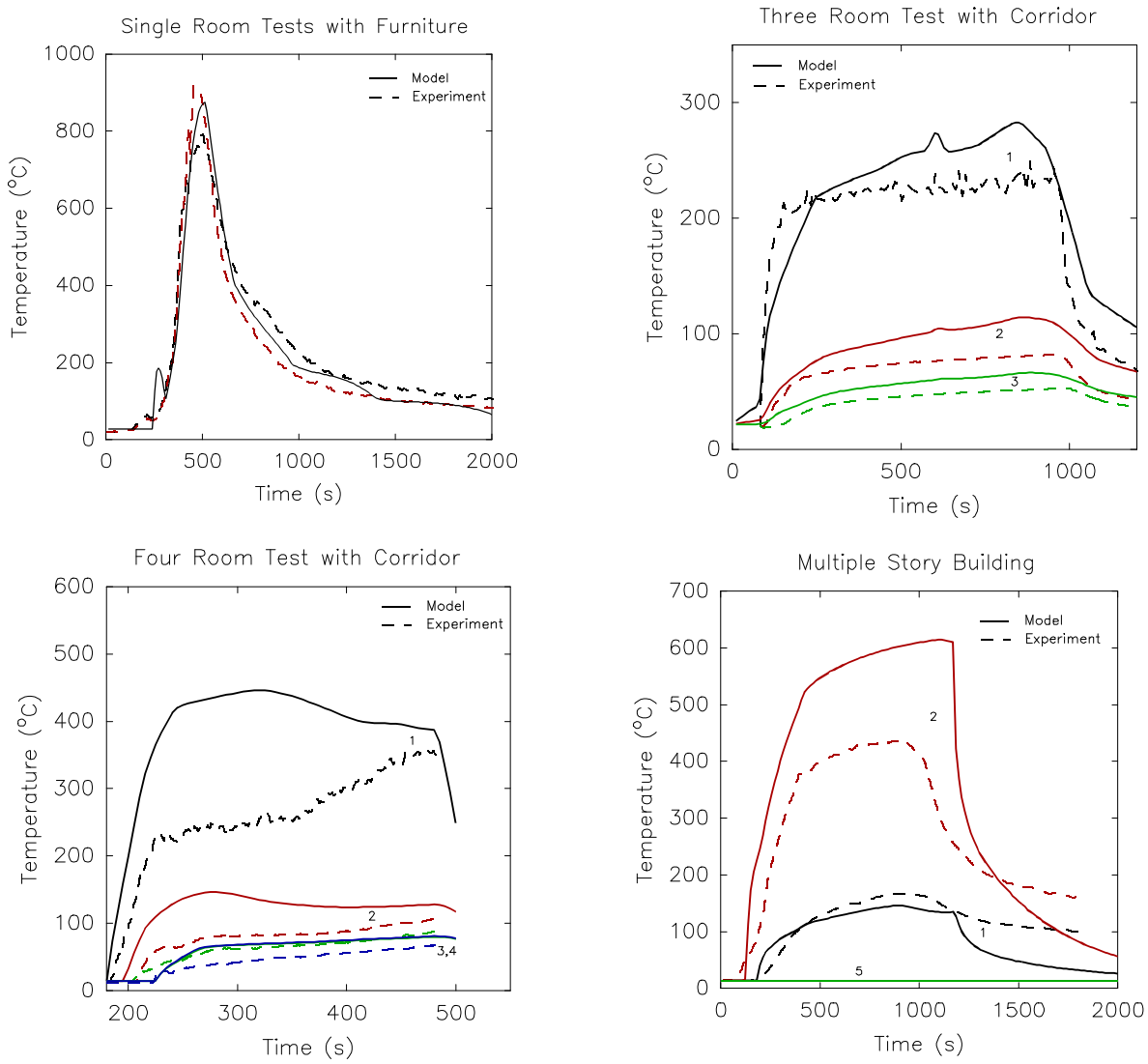


Figure 18. Comparison of measured and predicted upper layer temperatures for several tests. (Numbers indicate comparable rooms in the test structure.)

Systematic deviations exist for the remaining three data sets. Differences between model predictions and experimental measurements change monotonically over time (rising for the three-room test and falling for the four-rooms tests. Modelling of heat conduction (losing too much or too little heat to the surfaces) or lack of modelling of leakage (rooms are presumed perfectly sealed unless vents are included to simulate leakage) may account for the trends. The comparison of interface position for the four-room test with corridor seems an anomaly. Although a nearly closed space, the roughly level interface position from the experiment seems more typical of a test more open to the ambient. The model calculations would appear to better represent the mixing which would occur in a closed volume. Again, leakage may be a factor. With some leakage in the space, lower temperatures for both the lower and upper layer and higher (and more uniform) interface position would be calculated.

In general, upper layer temperature and interface position predicted by the model are somewhat higher than experimental measurements, with the differences ranging from -46 to 230 °C for the temperature and -0.19 to 1.5 m for the interface position. Conversely, the lower layer temperature is somewhat lower for the model than for the experiments (-60 to 5 °C). Presuming conservation of energy (an underlying assumption in *all* fire models), these three observations are consistent. A higher interface position gives rise to a smaller upper volume (and larger lower volume) within a room. With the same enthalpy in a smaller upper volume, higher temperatures result. This lends credence to the assumption of enthalpy conservation. Limitations inherent in the model also account partially for these trends. In the current version of CFAST, the lower layer is presumed to be clear. For the lower layer, energy is gained *only* by mixing or convection from surfaces. Adding radiative exchange to the lower layer would reduce the upper layer temperature and increase the lower layer temperature. Layer interface position is primarily affected by entrainment by the fire or at vents. Plume entrainment in CFAST is based on the work of McCaffrey [26] on circular plumes in relatively small spaces. For large fires in small spaces where the fire impinges on the ceiling (such as the single room tests with wall burning) or very small fires in large spaces (such as atria), these correlations may not be as valid.

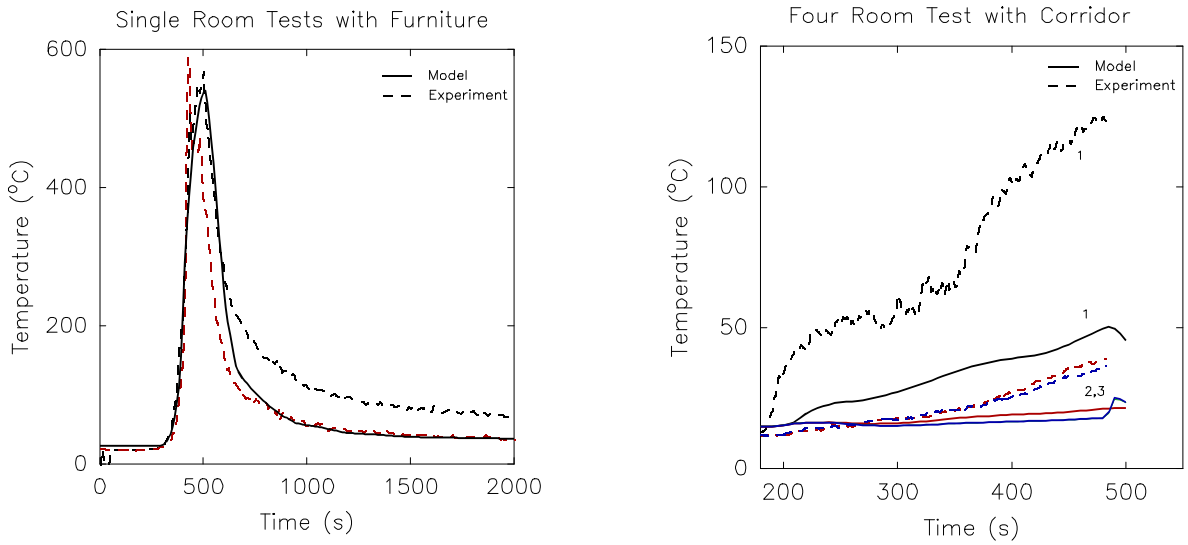


Figure 19. Comparison of measured and predicted lower layer temperatures for several tests. (Numbers indicate comparable rooms in the test structure.)

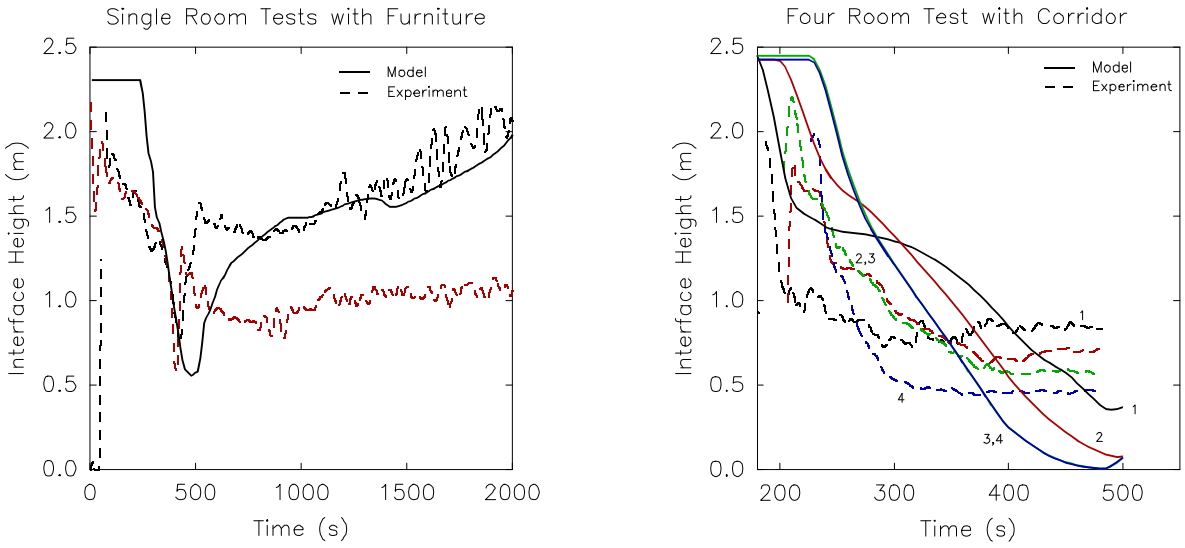


Figure 20. Comparison of measured and predicted layer interface position for several tests. (Numbers indicate comparable rooms in the test structure.)

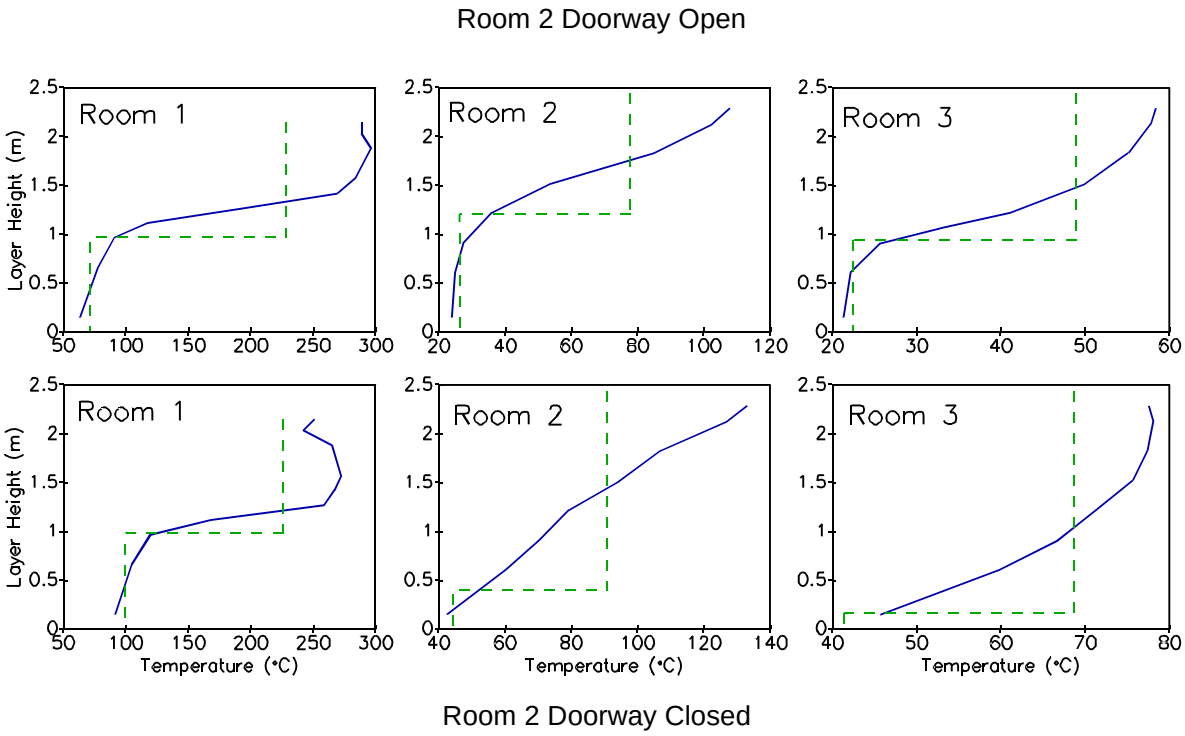


Figure 21. Average two-layer temperature and interface position calculated from experimentally-measured temperature profiles in a three-room experiment.

5.5.2 Gas Species

The fire chemistry scheme in CFAST is essentially a species balance from user-specified species yields and the oxygen available for combustion. Once generated, it is a matter of bookkeeping to track the mass of species throughout the various control volumes in a simulated building. It does, however, provide another check of the flow algorithms within the model. Since the major species (CO and CO_2) are generated only by the fire, the relative accuracy of the predicted values throughout multiple rooms of a structure should be comparable. Figure 22 and Table 8 show measured and predicted concentrations of O_2, CO_2, and CO in two of the tests studied.

For the single-room tests with furniture, the predicted concentrations are lower than those measured experimentally (averaging 5 percent low). This is probably due to the treatment of oxygen limited burning. In CFAST, the burning rate simply decreases as the oxygen level decreases. A user specified lower limit determines the point below which burning will not take place. This parameter could be finessed to provide better agreement with the experiment. For the present comparisons, it was always left at the default value.

For the four room test with corridor, the asymptotic values of the gas concentrations agree quite well. At first glance, the model predictions reach this equilibrium more quickly. An appreciation of the differences between the modeled parameters and the experimental measurements put this in perspective. From Figure 20, it takes about 100 s for the upper layer to descend to the level of the gas sampling port in the test. In addition, it is assumed than this point measurement is the bulk concentration of the entire upper layer. In reality, some vertical distribution not unlike the temperature profile (Figure 21) exists for the gas concentration as well. Since this measurement point is near the lower edge of the upper layer for a significant time, it should underestimate the bulk concentration until the layer is large in volume and well mixed.

For the multiple-story building test, predicted values for CO_2, CO, and O_2 are far lower than measured experimentally. Both the lower burning rate limit as well as leakage in the 100 year-old structure probably contribute to the differences between the experiments and model. In addition, values for species yields were simply literature values since no test data were available.

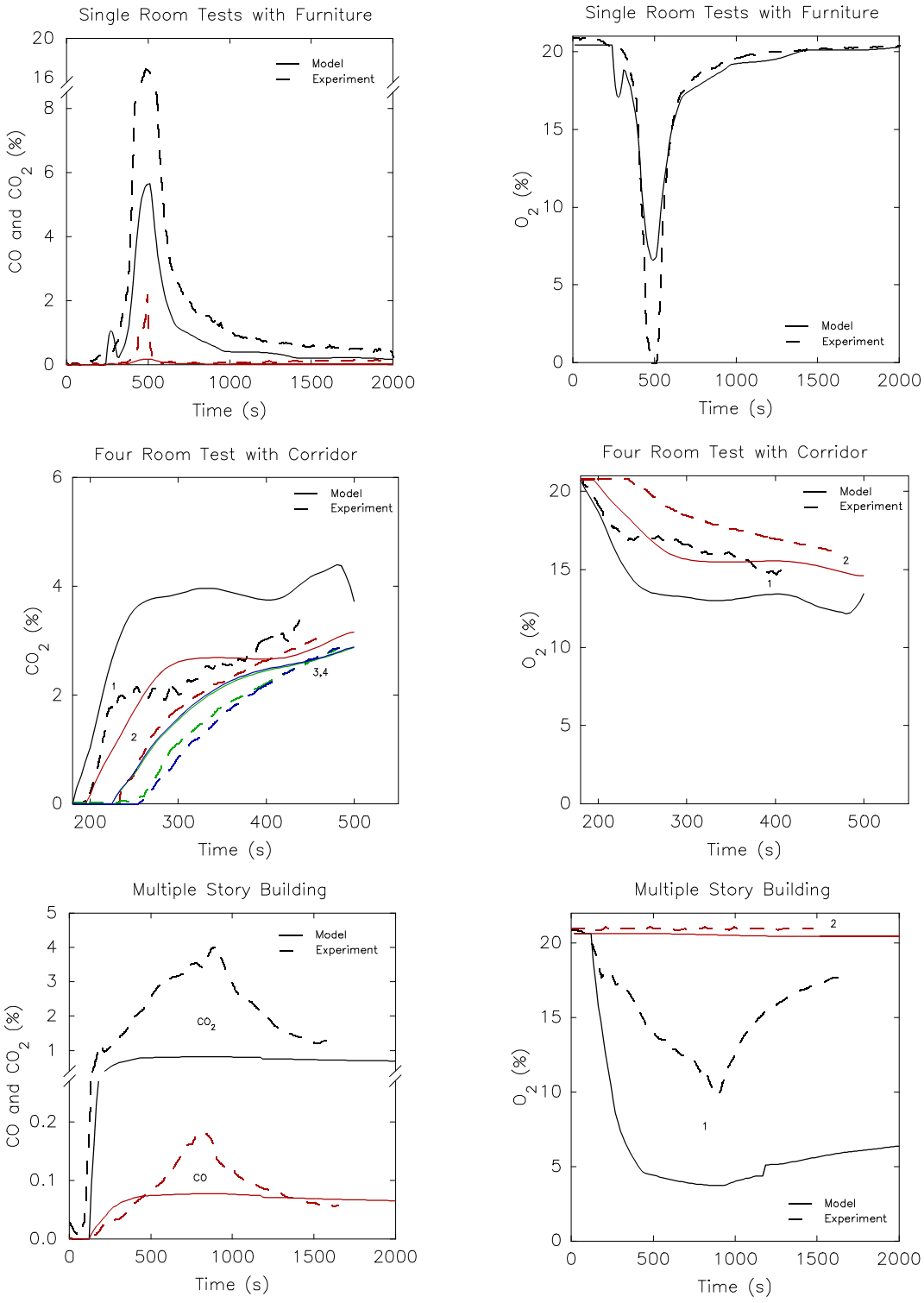


Figure 22. Comparison of measured and predicted gas species concentration for several tests. (Numbers indicate comparable rooms in the test structure.)

Oxygen Concentration

Numbers in parentheses are model predictions	Peak Value (%)	Time to Peak (s)	Steady-State Value (%)	Similar Shape?
Single-room furniture fire tests	0.01 (6.1)	510 (490)	--[a]	✓
	6.9 (10.2)	490 (510)	--	✓
Four-room tests with corridor[b]	--	--	17.9 (12.5) 18.0 (16.4)	✓
	--	--	16.1 (14.0) 18.1 (16.5)	✓
Multiple-story building test[b]	--	--	15.5 (2.9) 20.9 (20.4)	

Carbon Dioxide Concentration

Single-room furniture fire tests	17.0 (6.0)	480 (510)	--[a]	✓
	10.6 (4.2)	490 (510)	--	✓
Four-room tests with corridor[b]	--	--	2.3 (4.3)	✓
	--	--	2.4 (4.3)	✓
Multiple-story building test[b]	--	--	2.0 (0.9)	

Carbon Monoxide Concentration

Single-room furniture fire tests	2.2 (0.2)	490 (510)	--[a]	✓
	0.6 (0.1)	440 (510)	--	✓
Multiple-story building test[b]	--	--	0.8 (0.8)	

[a] not appropriate for the test.
[b] multiple entries indicate comparable rooms in the test structure.

Table 8. Comparison of experimental measurements and model predictions of oxygen concentration for several tests

5.5.3 Heat Release and Fire Pyrolysis Rate

Heat release rate and its intimately related fire pyrolysis rate are key indicators of fire hazard [71]. Peak values and time to reach peak values are typical scalar estimates used to represent the time-variant heat release rate and fire pyrolysis rate. For the single-room tests with furniture or wall-burning, these are appropriate indicators to judge the comparisons between model and experiment. For the three- and four-room tests with corridor or the multiple-story building tests, a steady state average is more appropriate.

Table 9 and Figure 23 compare measured and predicted heat release rates for the tests. In the CFAST model, the fire is specified as a series of straight line segments describing the pyrolysis rate, heat release rate, and species yields. Thus, the model predictions could be expected to agree quite well with experimental measurements. For tests where experimental data were available, the agreement is, not surprisingly, excellent – usually within 5 percent of the peak experimental values. Since this effectively just shows how well a series of line segments reproduces experimental measurement, this level of agreement is expected.

Numbers in parentheses are model predictions	Peak Value (kW)	Time to Peak (s)	Steady-State Value (kW)	Similar Shape?
Single-room furniture fire tests	2450 (2200)	480 (480)	--[a]	✓
	2600 (2350)	500 (510)	--	✓
Single-room tests with wall-burning	2050 (2000)	230 (200)	--	✓
	4000 (3150)	420 (370)	--	✓
Three-room test with corridor	--	--	86 (87)	✓
Four-room tests with corridor	--	--	n.r.[b]	✓
	--	--	n.r	✓
Multiple-story building test	--	--	n.r	✓

[a] not appropriate for the test.
[b] not available from experimental data.

Table 9. Comparison of experimental measurements and model predictions of heat release rate for several tests

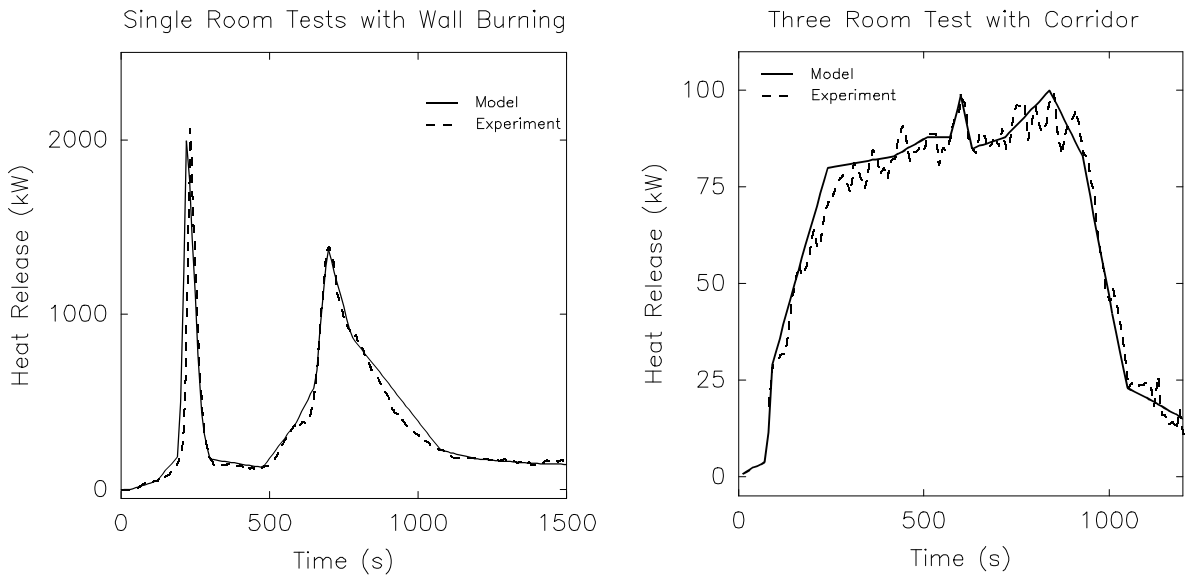


Figure 23. Comparison of measured and predicted heat release rates for two selected tests.

Times to peak values are always close. For two tests (the single-room with furniture and wall burning and the multiple-story building), the heat release rate in the room is limited by the available oxygen. Additional burning outside the room (seen in the single-room with furniture) accounts for the remainder of the heat released.

For the three-room test with corridor, multiple replicate tests put the agreement between the model and experiments in perspective. For all tests in the original study [18], the coefficients of variation (the standard deviation expressed as a percentage of the mean) ranged from 4 to 52 percent. In another study, precision to within 15 percent for fires of 2.5 MW was noted [67]. Thus, the simplification of specifying the fire growth as a series of straight lines is easily justified with the expected accuracy of experimental measurements.

For the multiple-story building test, *no* pyrolysis rate or heat release rate data were available. Estimates of the "steady-state" burning rate, time to reach "steady-state," and duration of "steady-state" burning were made from available correlations for wood cribs [72], [73]. Although the comparisons for this test should be considered approximate, it was included since, if successful, the scope of the model is extended considerably to a large multiple-story building with mechanical ventilation.

5.5.4 Pressure

The differential pressure across an opening drives the flow through the opening. For each room, the CFAST model calculates a differential pressure at floor level, referenced to ambient. Noting that the ambient pressure is approximately 100 kPa, typical pressure drops across openings induced by fires are but a small fraction of the ambient pressure – typically from less than 1 Pa to perhaps a few hundred Pascals in well-sealed enclosures. The ability to model these extremely small differential pressures provides another check on the flow algorithms in the model. These are, however, expected to be difficult to model and measure accurately. Thus, agreement within a few pascals is often considered acceptable. In four of the five experimental test series, measurements (corrected to floor level) were available which could be compared to these predicted values (measurements were not available for the single room tests with furniture).

Figure 24 and Table 10 show the comparisons. For most cases, the agreement is reasonable, with the difference between measured and predicted values typically less than 2 Pa and for some experiments, less than 0.5 Pa. Trends displayed in the experimental data are replicated by the model predictions. Some interesting exceptions are apparent however. In major part, these are due to quantities unknown in the experiments (leakage). Not all of the onus for agreement should be placed on the model, however. Only one of the test series included any estimate of leakage through cracks in the buildings. Logically, unless directed otherwise, the model assumes *no* leakage from any room. This leakage can have a dramatic effect on the results predicted by the model. Figure 25 illustrates the effect of leakage for a single room with a single doorway and an upholstered chair used as the fire source. Leakage areas from 0 to 100 percent of the vent area were simulated with a second vent of appropriate size and placed at floor level (much of the leakage in rooms take place at floor level). Both temperatures and pressures are seen to change by more than a factor of two (other variables can be expected to change with similar variation). Temperatures changes by about 20 percent with only a 10 percent leakage area. The effect on pressure is not quite as straightforward, but for larger leakages changes in concert with the temperature. For the four-room tests with corridor, leakage from the "well sealed rooms" was estimated via measurement at not more than 25 percent of the total vent area.

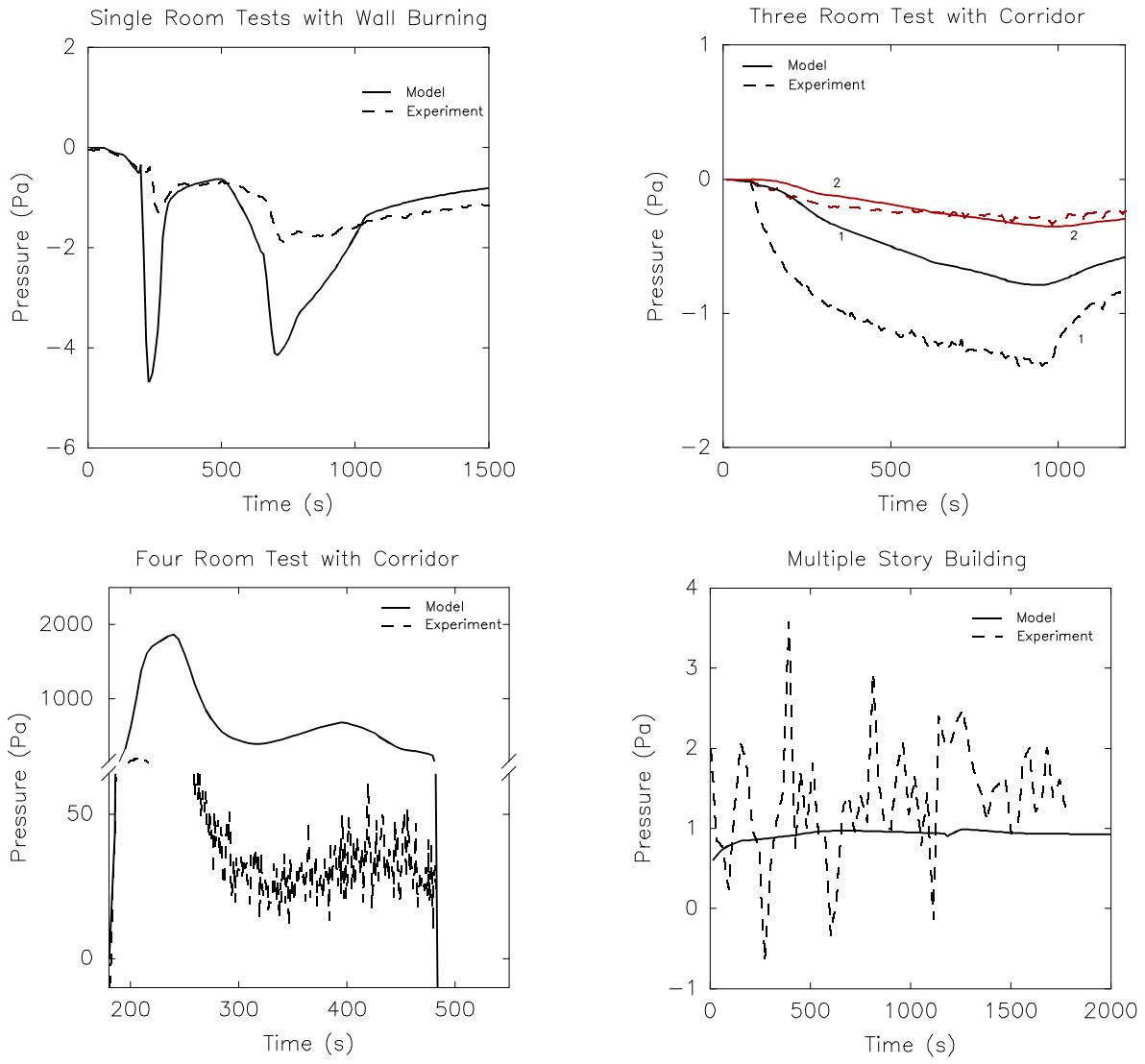


Figure 24. Comparison of measured and predicted pressures for several tests.
(Numbers indicate comparable rooms in the test structure.)

Numbers in parentheses are model predictions	Peak Value	Time to Peak	Steady-State Value	Similar Shape?
Single-room tests with wall-burning	-1.9 (-4.6)	730 (750)	--	✓
	-1.9 (-5.6)	520 (490)	--	✓
Three-room test with corridor	--	--	-1.1 (-0.6) -0.2 (-0.5)	✓
Four-room tests with corridor	--	--	-1.0 (-2.1)	✓
	--	--	36 (22)	
Multiple-story building test	--	--	2.4 (1.3)	✓

[a] not appropriate for the test.

Table 10. Comparison of experimental measurements and model predictions of room pressure for several tests

5.5.5 Flow Through Openings

In the control volume approach, the differential form of the momentum equation for the zones is not solved directly. Rather, the momentum transfer at the zone boundaries is included by using Bernoulli's approximation for the velocity equation. This solution is augmented for restricted openings by using flow coefficients [31], [74] to allow for constriction in vents. The flow coefficients allow for an effective constriction of fluid flow which occurs for vents with sharp edges. In CFAST, these coefficients are for rectangular openings in walls whose surfaces are much larger than the opening.

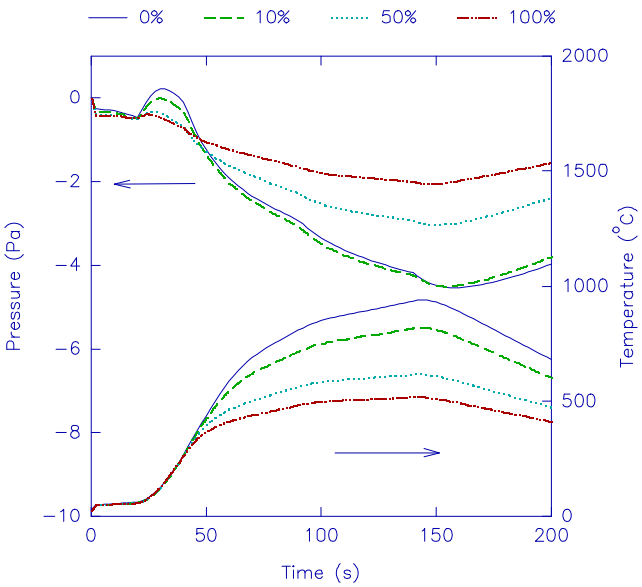


Figure 25. Effect of leakage on calculated temperatures and pressures in an arbitrary single-room fire.

Figure 26 and Table 11 compare measured and predicted mass flows through doorways in two of the tests studied. For the three-room test with corridor, flow through two doorways of the same test are shown (one between the fire room and the corridor and one between the corridor and the outdoors). Not surprisingly, the flow is typically somewhat underpredicted by the model (from -0.14 to -0.58 kg/s). The vent flow in CFAST includes mixing phenomena at the vents. As hot gases from one compartment leave that compartment and flow into an adjacent compartment, a door jet can exist which is analogous to a normal fire plume, but with an extended

flat plume similar to a waterfall. This places its use outside the normal range of the plume model [26] and perhaps beyond its range of validity. However, no reliable correlation yet exists for the extended flat plume which occurs in vent flow. Examining the trends of prediction of upper layer temperature in tests with multiple rooms (Tables 1 and 2), the typical over-prediction in the room of fire origin is far greater than for other rooms in the structures. The under-prediction of the mass flows probably accounts for this as a cascading effect as you move away from the room of fire origin.

Numbers in parentheses are model predictions	Peak Value	Time to Peak	Steady-State Value	Similar Shape?
Single-room furniture fire tests	1.2 (1.3)	380 (410)	--[a]	✓
	1.9 (1.9)	560 (460)	--	✓
Three-room test with corridor	--	--	0.4 (0.3)	✓

[a] not appropriate for the test.

Table 11. Comparison of experimental measurements and model predictions of mass flow through openings for several tests

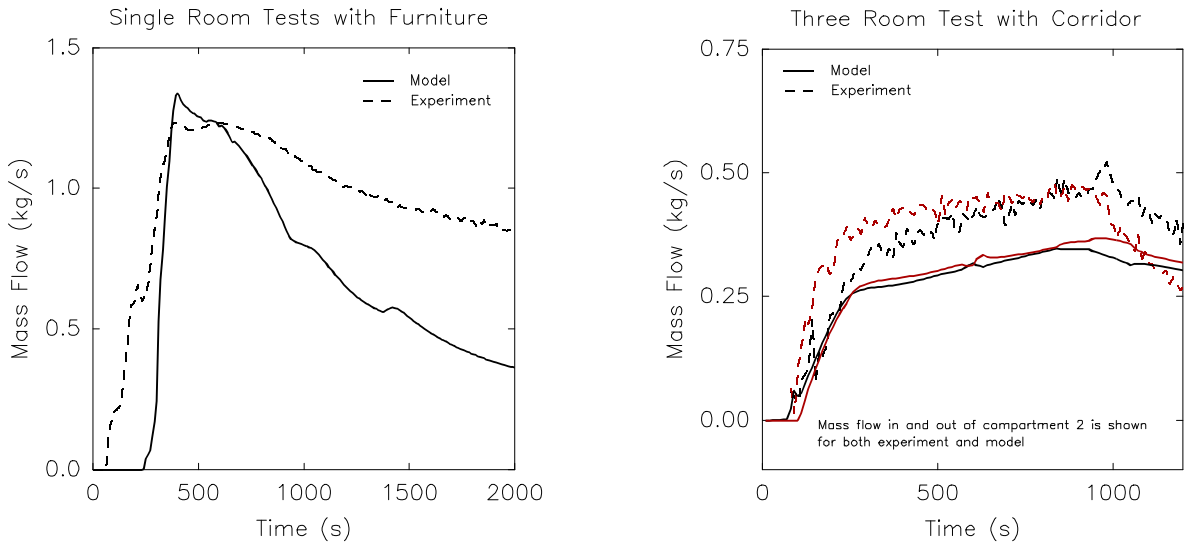


Figure 26. Comparison of measured and predicted mass flow through vents for several tests. (Numbers indicate comparable rooms in the test structure.)

6 Description of the Data File

The CFAST model requires a description of the problem to be solved. This section provides a description for the input data used by the model. In general, the order of the data is not important. The one exception to this is the first line which specifies the version number and gives the data file a title.

Most entries in the input data file can be generated using CEdit. CEdit provides online help information in addition to context-sensitive error checking. For example, CEdit will not allow the user to select a fire compartment outside the range of compartments specified on the Geometry screen. Because of these features, CEdit is the preferred method for creating and editing most CFAST input files. However, some input file key words are not supported by CEdit. For these special cases, editing of the ASCII input file using any ASCII text editor is necessary. The following sections detail the available input file key words and group them by their availability within CEdit. Subsection titles for the CEdit key words correspond to the subsection titles in the CEdit chapter of this reference. This has been done as an aid to understanding the organization of the input file.

The number of lines in a given data set will vary depending, for example, on the number of openings or the number of species tracked. A number of parameters such as heat transfer and flow coefficients have been set within CFAST as constants. Please refer to the section on source terms for the values of these parameters.

6.1 General Format of an Input File Line

Each line of the data file begins with a key word which identifies the type of data on the line. The key words currently available are:

CEILI	specify name of ceiling descriptor(s)	(N)
CHEMI	miscellaneous parameters for kinetics	(7)
CJET	ceiling jet	(1)
CO	CO/CO_2 mass ratio	(lfmax)
CT	fraction of fuel which is toxic	(lfmax)
CVENT	opening/closing parameter	(lfmax + 3)
DEPTH	depth of compartments	(N)
DUMPR	specify a file name for saving time histories	(1)
EAMB	external ambient	(3)
FAREA	area of the base of the fire	(lfmax)
FHIGH	height of the base of the fire	(lfmax)
FLOOR	specify the name of floor property descriptor(s)	(N)
FMASS	pyrolysis rate	(lfmax)
FPOS	exact position of the fire using x, y, z coordinates	(3)
FQDOT	heat release rate	(lfmax)
FTIME	points of time on the fire timeline	(DETERMINED BY MAXIMUM LINE SIZE)
HCL	hcl/pyrolysis mass ratio	(lfmax)
HCN	hcn/pyrolysis mass ratio	(lfmax)
HCR	hydrogen/carbon mass ratio of the fuel	(lfmax)
HEIGH	interior height of a compartment	(N)
HI/F	absolute height of the floor of a compartment	(N)

HVENT	specify vent which connect compartments horizontally	(7)
INELV	specify **interior** node elevations (for ventilation ducts)	(2 x # of interior nodes)
LFBO	compartment of fire origin	(1)
LFBT	type of fire	(1)
LFMAX	**NO LONGER SUPPORTED IN THE DATA FILE**	
LFPOS	position of the fire in the compartment	(1)
MVDCT	describe a piece of (circular) duct work	(9)
MVFAN	give the pressure - flow relationship for a fan	(5 to 9)
MVOPN	Specify an opening between a compartment and ventilation system	(5)
OBJECT	additional objects to be burned	(7)
OBJFL	alternative object database file	(1)
OD	C/CO_2 mass ratio	(lfmax)
O2	ratio of oxygen to carbon in the fuel	(lfmax)
RESTR	specify a restart file	(2)
TAMB	ambient inside the structure	(3)
THRMF	alternative thermal properties file	(1)
TIMES	time step control of the output	(5)
VERSN	version number and title	(fixed format 2)
VVENT	specify a vent which connects compartments vertically	(4)
WALLS	specify the name of wall property descriptor(s)	(N)
WIDTH	width of the compartments	(N)
WIND	scaling rule for wind effects	(3)

The number in parenthesis is the maximum number of entries for that line. “N” represents the number of compartments being modeled. The outside (ambient) is designated by one more than the number of compartments, N+1. Thus, a three compartment model would refer to the outside as compartment four. An entry for LFMAX is no longer supported directly. The value for LFMAX is determined by the number of entries on the FTIME line.

Each line of input consists of a label followed by one or more alphanumeric parameters associated with that input label. The label must always begin in the first space of the line and be in capital letters. Following the label, the values may start in any column, and all values must be separated by either a comma or a space. Values may contain decimal points if needed or desired. They are not required. Units are standard SI units. Most parameters have default values which can be utilized by omitting the appropriate line. These are indicated in the discussion. The maximum line length is 128 characters, so all data for each key word must fit in this number of characters.

6.2 Key Words Available in CEdit

6.2.1 Overview

The first line in the file must be the version identification. It is a required input. The VERSN line is the line that CFAST keys on to determine whether it has a correct data file. The format is fixed, that is the data must appear in the columns specified in the text. A TIMES line is also required in order to specify the length of time over which the simulation takes place. The RESTR line is an optional line used to restart the model at a specified simulation time within an existing history file.

Example:

```
VERSN    1 Example Case for CFAST 1.6 User's Guide
TIMES    200    10    10     0     0
```

Key word: RESTR Input: Restart File (See Section 6.2.10), Restart Time	
Restart Time (s)	A time step is given after the name of the file and specifies at what time the restart should occur.

Key word: VERSN Inputs: Version Number, Title	
Version Number	The version number parameter specifies the version of CFAST for which the input data file was prepared. Normally, this would be 1. It must be in columns 8-9.
Title	The title is optional and may consist of letters, numbers, and/or symbols that start in column 11 and may be up to 50 characters. It permits the user to uniquely label each run.

Key word: TIMES Inputs: Simulation time, Print Interval, History Interval, Display Interval, Copy Count	
Simulation Time (s)	Simulation time is the length of time over which the simulation takes place. The maximum value for this input is 86400 s (1 day). The simulation time parameter is required.
Print Interval (s)	The print interval is the time interval between each printing of the output values. If omitted or less than or equal to zero, no printing of the output values will occur.
History Interval (s)	The history interval is the time interval between each writing of the output to the history file. The history file stores all of the output of the model at the specified interval in a format which can be efficiently retrieved for use by other programs. Section 6.2.10 provides details of the history file. A zero must be used if no history file is to be used.
Display Interval (s)	The display interval is the time interval between each graphical display of the output as specified in the graphics specification, section 6.3.2. If omitted, no graphical display will occur. There is a maximum of 200 intervals allowed. If the choice for this parameter would yield more than 200 writes, the graphs are truncated to the first 200 points.
Copy Count	Copy count is the number of copies of each graphical display to be made on the selected hard copy device as specified in the graphics specification, section 6.3.2. If omitted, a value of zero (no copies) is assumed.

6.2.2 Ambient Conditions

The ambient conditions section of the input data allows the user to specify the temperature, pressure, and station elevation of the ambient atmosphere, as well as the absolute wind pressure to which the structure is subjected. There is an ambient for the interior and for the exterior of the structure. The key word for the interior of the structure is TAMB and for the exterior of the structure is EAMB. The form is the same for both. The key word for the wind information is WIND. The wind modification is applied only to the vents which lead to the exterior. Pressure interior to a structure is calculated simply as a lapse rate based on the NOAA tables [75]. For the exterior, the nominal pressure is modified by:

$$\delta(p) = C_w \rho V^2 \qquad \text{where} \qquad V = V_w \left(\frac{H_i}{H_w} \right)^{p_w} . \tag{90}$$

This modification is applied to the vents which lead to the exterior ambient. The pressure change calculated above is modified by the wind coefficient for each vent. This coefficient, which can vary from -1.0 to +1.0, nominally from -0.8 to +0.8, determines whether the vent is facing away from or into the wind. The pressure change is multiplied by the vent wind coefficient and added to the external ambient for each vent which is connected to the outside.

The choice for the station elevation, temperature and pressure must be consistent. Outside of that limitation, the choice is arbitrary. It is often convenient to choose the base of a structure to be at zero height and then

reference the height of the structure with respect to that height. The temperature and pressure must then be measured at that position. Another possible choice would be the pressure and temperature at sea level, with the structure elevations then given with respect to mean sea level. This is also acceptable, but somewhat more tedious in specifying the construction of a structure. Either of these choices works though because consistent data for temperature and pressure are available from the Weather Service for either case.

If the EAMB or TAMB line is not included in the input file, the default values specified below are used. The WIND line is optional.

Example:

```
TAMB   300.  101300.      0.
EAMB   300.  101300.      0.
```

Key words: EAMB and TAMB Inputs: Ambient Temperature, Ambient Pressure, Station Elevation (External and Internal, respectively)	
Ambient Temperature (K)	Ambient temperature is the temperature of the ambient atmosphere. Default is 300.
Ambient Pressure (Pa)	The ambient pressure is the pressure of the ambient atmosphere. Default is 101300.
Station Elevation (m)	The station elevation is the elevation of the point at which the ambient pressure and temperature (see above) are measured. The reference point for the elevation, pressure and temperature must be consistent. This is the reference datum for calculating the density of the atmosphere as well as the temperature and pressure inside and outside of the structure as a function of height. Default is 0.

Key word: WIND Inputs: Wind Speed, Reference Height, Lapse Rate Coefficient	
Wind Speed (m/s)	Wind speed at the reference elevation. The default is 0.
Reference Height (m)	Height at which the reference wind speed is measured. The default is 10 m.
Lapse Rate Coefficient	The power law used to calculate the wind speed as a function of height. The default is 0.16.

6.2.3 Geometry

This section allows the user to portray the geometry of the structure being modeled. The size and location of every compartment in the structure MUST be described. The maximum number of compartments is 15 compartments (plus the outdoors). The structure of the data is such that the compartments are described as entities, and then connected in appropriate ways. It is thus possible to have a set of compartments which can be configured in a variety of ways. In order to specify the geometry of a structure, it is necessary to give the physical characteristics. Thus the lines labelled HI/F, WIDTH, DEPTH and HEIGH are all required. Each of these lines requires "N" data entries, that is one for each compartment.

Example:

```
WIDTH    4.00   4.00
DEPTH    4.00   4.00
HEIGH    2.30   2.30
HI/F     0.00   2.30
```

Key word: WIDTH Input: Compartment Width	
Compartment Width (m)	Compartment width specifies the width of the compartment. The number of values on the line must equal the number of compartments in the simulation.

Key word: DEPTH Input: Compartment Depth	
Compartment Depth (m)	Compartment depth specifies the depth of the compartment. The number of values on the line must equal the number of compartments in the simulation.

Key word: HEIGH Input: Compartment Height	
Compartment Height (m)	Compartment Height specifies the height of the compartment. The number of values on the line must equal the number of compartments in the simulation.

Key word: HI/F Input: Floor Height	
Floor Height (m)	The floor height is the height of the floor of each compartment with respect to station elevation specified by the TAMB parameter. The reference point must be the same for all elevations in the input data. The number of values on the line must equal the number of compartments in the simulation.

6.2.4 Vents(doors,...)

The Vents(doors,...) section of the input data file is required to specify horizontal flow connections between compartments in the structure. These may include doors between compartments or windows in the compartments (between compartments or to the outdoors). These specifications do **not** correspond to physically connecting the walls between specified compartments. Lack of an opening prevents flow. Openings to the outside are included as openings to a compartment with a number one greater than the number of compartments described in the Geometry section. The key word is HVENT. If the HVENT line is entered, the first six entries on the line are required. There is an optional seventh parameter to specify a wind coefficient. The soffit and sill specifications are with respect to the first compartment specified and is not symmetric. Reversing the order of the compartment designations does make a difference.

Horizontal flow vents may be opened or closed during the fire with the use of the CVENT key word. The initial format of CVENT is similar to HVENT specifying the connecting compartments and vent number. Each CVENT line in the input file details the open/close time dependent characteristics for one horizontal flow vent by specifying a fractional value for each LFMAX time entry. The default is 1.0 which is a fully open vent. A value of 0.5 would specify a vent which was halfway open.

Example:

```
HVENT   1  3  1   1.07   2.00   0.00   0.00
HVENT   2  3  1   1.07   2.00   1.00   0.00
CVENT   1  3  1   1.00   1.00
CVENT   2  3  1   1.00   1.00
```

Key word: CVENT Inputs: First Compartment, Second Compartment, Vent Number, Width	
First Compartment	The first compartment.
Second Compartment	The second compartment is the compartment number to which the first compartment is connected.
Vent Number	This number specifies which vent is being described. It can range from one to four.
Width	Fraction that the vent is open. This applies to the width only. The sill and soffit are not changed. The number of values on the line must equal the number of points on the fire timeline.

Key word: HVENT Inputs: First Compartment, Second Compartment, Vent Number, Width, Soffit, Sill, Wind	
First Compartment	The first compartment is simply the first connection.
Second Compartment	The second compartment is the compartment number to which the first compartment is connected. The order has one significance. The height of the sill and soffit are with respect to the first compartment specified.
Vent Number	There can be as many as four vents between any two compartments. This number specifies which vent is being described. It can range from one to four.
Width (m)	The width of the opening.
Soffit (m)	Position of the top of the opening above the floor of the compartment number specified as the first compartment.
Sill (m)	Sill height is the height of the bottom of the opening above the floor of the compartment number specified as the first compartment.
Wind	The wind coefficient is the cosine of the angle between the wind vector and the vent opening. This applies only to vents which connect to the outside ambient (specified with EAMB). The range of values is -1.0 to +1.0. If omitted, the value defaults to zero.

6.2.5 Vents(ceiling,...)

The Vents(ceiling,...) section of the input data file describes any vertical flow openings, such as scuddles, between compartments in the structure (or between a compartment and the outdoors). Openings to the outside are included as openings to a compartment with a number one greater than the number of compartments described in the Geometry section. Each VVENT line in the input file describes one vertical vent. There are four parameters, the connected compartments, the shape of the opening, and the effective area of the vent. At the present time, there is not an equivalent CVENT mechanism for opening or closing the vertical vents.

Example:

```
VVENT   2   1  1.00   1
```

Key word: VVENT Inputs: First Compartment, Second Compartment, Area, Shape	
First Compartment	The first compartment is simply the first connection.
Second Compartment	The second compartment is the compartment number to which the first compartment is connected.
Area (m^2)	This is the effective area of the opening. For a hole, it would be the actual opening. For a diffuser, the effective area is somewhat less than the geometrical size of the opening.
Shape	1 for circle or 2 for square.

6.2.6 Fans, Ducts,...

These key words are used to describe a mechanical ventilation system. The MVOPN line is used to connect a compartment to a node in the mechanical ventilation system. The elevation for each of these exterior nodes is specified as a relative height to the compartment floor on the MVOPN line. The MVDCT key word is used to specify a piece of the mechanical ventilation duct work. CAUTION: Nodes specified by each MVDCT entry must connect with other nodes, fans, or compartments. Do not specify ducts which are isolated from the rest of the system. Specify interior elevations of the mechanical ventilation nodes using the INELV line. All node elevations can be specified, but elevations for the exterior nodes, that is those connected to a compartment, are ignored. These heights are determined by entries on the MVOPN line. The heights for interior nodes are absolute heights above the reference datum specified by TAMB. The heights are specified in pairs with the node number followed by the height.

A fan is defined using the MVFAN line to indicate node numbers and to specify the fan curve with power law coefficients. There must be at least one and a maximum of five coefficients specified for each MVFAN entry. The fan coefficients are simply the coefficients of an interpolating polynomial for the flow speed as a function of the pressure across the fan housing. In this example, the coefficients:

```
  B( 1) =    0.140E+00      b(1)
  B( 2) =   -0.433E-03      b(2) x p
```

were calculated from entries made in CEdit:

```
             PRESSURE          FLOW

Minimum          0.00        0.1400
Maximum        300.00        0.0101
```

Example:

```
MVOPN   1   1 V   2.10   0.12
MVOPN   2   3 V   2.10   0.12
MVDCT   1   2   2.30   0.10 .00200   0.00 1.0000   0.00 1.0000
MVFAN   2   3   0.00 300.00  0.140E+00 -0.433E-03
INELV   1   2.10   2   4.40   3   4.40
```

Key word: INELV Inputs: Node Number, Height	
Node Number	Number of an interior node.
Height (m)	Height of the node with respect to the height of the reference datum, specified by TAMB or EAMB.

Key word: MVDCT Inputs: First Node Number, Second Node Number, Length, Diameter, Absolute Roughness, First Flow Coefficient, First Area, Second Flow Coefficient, Second Area	
First Node Number	First node number. This is a node in the mechanical ventilation scheme, not a compartment number (see MVOPN).
Second Node Number	Second node number.
Length (m)	Length of the duct.
Diameter (m)	All duct work is assumed to be circular. Other shapes must be approximated by changing the flow coefficient. This is done implicitly by network models of mechanical ventilation and forced flow, but must be done explicitly here.
Absolute Roughness (m)	Roughness of the duct.
First Flow Coefficient	Flow coefficient to allow for an expansion or contraction at the end of the duct which is connected to node number one. To use a straight through connection (no expansion or contraction) set to zero.
First Area (m^2)	Area of the expanded joint.
Second Flow Coefficient	Coefficient for second node.
Second Area (m^2)	Area at the second node.

Key word: MVFAN Inputs: First Node, Second Node, Minimum Pressure, Maximum Pressure, Coefficients	
First Node	First node in the mechanical ventilation system to which the fan is connected.
Second Node	Second node to which the fan is connected.
Minimum Pressure (Pa)	Lowest pressure of the fan curve. Below this value, the flow is assumed to be constant.
Maximum Pressure (Pa)	Highest pressure at which the fan will operate. Above this point, the flow is assumed to stop.
Coefficients	At least one, and a maximum of five coefficients, to specify the flow as a function of pressure.

Key word: MVOPN Inputs: Compartment Number, Duct Work Node Number, Orientation, Height, Area	
Compartment Number	Specify the compartment number.
Duct Work Node Number	Corresponding node in the mechanical ventilation system to which the compartment is to be connected.
Orientation	V for vertical or H for horizontal.
Height (m)	Height of the duct opening above the floor of the compartment.
Area (m^2)	Area of the opening into the compartment.

6.2.7 Thermal Properties

The thermophysical properties of the enclosing surfaces are described by specifying the thermal conductivity, specific heat, emissivity, density, and thickness of the enclosing surfaces for each compartment. Currently, thermal properties for materials are read from a thermal database file unique to CFAST. The data in the file simply gives a name (such as CONCRETE) which is a pointer to the properties in the thermal database. The thermophysical properties are specified at *one* condition of temperature, humidity, etc. There can be as many as three layers per boundary, but they are specified in the thermal database itself.

If the thermophysical properties of the enclosing surfaces are not included, CFAST will treat them as adiabatic (no heat transfer). If a name is used which is not in the database, CEdit will turn off the conduction calculation, and CFAST will stop with an appropriate error message.

Since most of the heat conduction is through the ceiling, and since the conduction calculation takes a significant fraction of the computation time, it is recommended that initial calculations be made using the ceiling only. Adding the walls generally has a small effect on the results, and the floor contribution is usually negligible. Clearly, there are cases where the above generalization does not hold, but it may prove to be a useful screening technique.

The default name for the thermal properties database is THERMAL.TPF. Another name can be used by selecting it during installation, or by using the key word THRMF in the CFAST data file.

Example:

```
CEILI GYPSUM    GYPSUM
WALLS PINEWOOD  PINEWOOD
FLOOR CONCRETE  CONCRETE
```

Key word: CEILI Inputs: Ceiling Materials	
Ceiling Materials	The label CEILI indicates that the names of thermophysical properties on this line describe the ceiling material. If this parameter is present, there must be an entry for each compartment.

Key word: WALLS Inputs: Wall Materials	
Wall Materials	The label WALLS indicates that the names of thermophysical properties on this line describe the wall material. If this parameter is present, there must be an entry for each compartment.

Key word: FLOOR Inputs: Floor Materials	
Floor Materials	The label FLOOR indicates that the names of thermophysical properties on this line describe the floor material. If this parameter is present, there must be an entry for each compartment.

Key word: THRMF Input: Thermal Database	
Thermal Database	The name specifies a file (up to 12 characters) from which the program reads thermophysical data. If this parameter is not specified, then either the default (THERMAL.DAT) is used, or the name is read from the configuration file.

6.2.8 Fire Specification

The fire specifications allow the user to describe the fire source in the simulation. The location and position of the fire are specified using the LFBO and FPOS lines. Chemical properties of the fuel are specified with the CHEMI key word along with miscellaneous parameters. Turn the ceiling jet calculations on by using the CJET key word. By default, the ceiling jet is not included in a CFAST simulation.

By default, the fire is placed in the center of the compartment on the floor (see Figure 27). To place the fire in a different location, the FPOS key word may be included in the input file. If values for any of the three variables are invalid (i.e., less than zero or greater than the compartment dimension in the appropriate direction), the location for that direction defaults to the center of the appropriate direction.

CFAST no longer supports use of the LFMAX key word in the data file. LFMAX is now determined by the number of entries on the FTIME line used to specify points of the fire timeline. The time dependent variables of the fire are described with a series of mass loss rate, rate of heat release, fuel height, and fuel area inputs. All of these specifications are optional. If entered, a total of LFMAX+1 values must be included for each time dependent input line. The defaults shown for each key word reflect the values for methane.

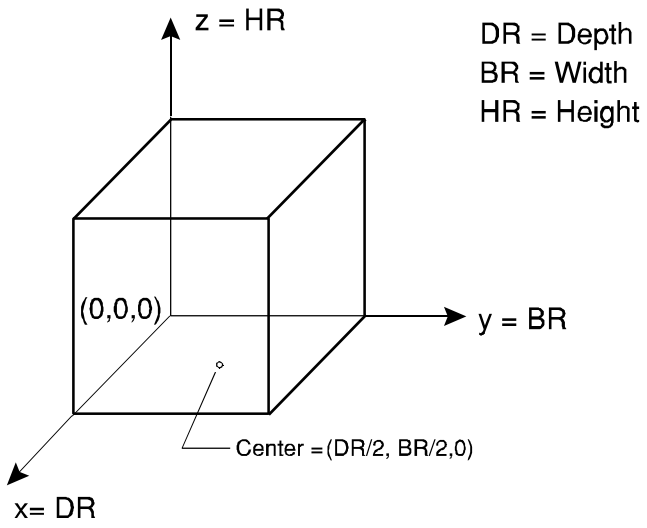


Figure 27. Coordinate system for determining fire position in a compartment.

With the three parameters, the heat of combustion (HOC) from CHEMI, FMASS and FQDOT, the pyrolysis and heat release rate are over specified. The model uses the last two of the three to obtain the third parameter. That is, if the three were specified in the order HOC, FMASS and FQDOT, then FQDOT would be divided by FMASS to obtain the HOC for each time interval. If the order were FMASS, FQDOT and HOC, then the pyrolysis rate would be determined by dividing the heat release rate by the heat of combustion. If only two of the three are given, then those two will determine the third, and finally, if none or only one of the parameters is present, the defaults shown are used.

Species production rates are specified in a manner similar to the fire, entering the rates as a series of points with respect to time. The species which are followed by CFAST are:

- Carbon Dioxide
- Carbon Monoxide
- Concentration-Time Product
- Hydrogen Cyanide
- Hydrogen Chloride
- Nitrogen
- Oxygen
- Soot (Smoke Density)
- Total Unburned Hydrocarbons
- Water

The program performs a linear interpolation between the time points to determine the time of interest.

For a type one (LFBT=1) fire, only the concentration-time product of pyrolysate (CT) can be specified. No other species are followed. For a type two (LFBT=2) fire, nitrogen (N2), oxygen (O2), carbon dioxide (CO2), carbon monoxide (CO), hydrogen cyanide (HCN), hydrogen chloride (HCL), soot (OD), unburned fuel and water (H2O) are followed. For a type two fire, HCN, HCL, CT, O2, OD, CO and the hydrogen to carbon ratio (HCR) can be specified. In all cases, the unit of the production rates is kg/kg. However, the meaning of the production rates is different for the several types of species. See the discussion for each species key word below for the meaning of the corresponding production rate.

Example:

```
CHEMI  16.    0.  10.0    18100000.  300. 400.   0.
LFBO   1
LFBT   2
FPOS   2.00   2.00   0.00
FTIME  400.
FMASS  0.0014 0.0014
FHIGH  0.00   0.01
FAREA  0.00   0.00
FQDOT  2.53E+04 2.53E+04
CJET   OFF
HCR    0.333 0.333
CO     0.010 0.010
```

Key word: CHEMI Inputs: Molar Weight, Relative Humidity, Lower Oxygen Limit, Heat of Combustion, Initial Fuel Temperature (see sec. 6.3.1), Gaseous Ignition Temperature (see sec. 6.3.1), Radiative Fraction (see sec. 6.3.1)	
Molar Weight	Molecular weight of the fuel vapor. This is the conversion factor from mass density to molecular density for "tuhc." Default is 16. It is used only for conversion to ppm, and has no effect on the model itself.
Relative Humidity (%)	The initial relative humidity in the system. This is converted to kilograms of water per cubic meter [76].
Lower Oxygen Limit (%)	The limit on the ratio of oxygen to other gases in the system below which a flame will not burn. This is applicable only to type (LFBT) 2 or later fires. The default is 10.
Heat of Combustion (J/kg)	Heat of combustion of the fuel. Default is 50000000.

Key word: CJET Input: OFF, CEILING, WALL, or ALL	
Current Setting	To include the calculation for the ceiling, wall, or both surfaces, the CJET key word is used together with one of the identifiers CEILING, WALL, or ALL. For example, to turn the ceiling on, use "CJET CEILING." At present, this key word effects only the calculation of the convective heating boundary condition for the conduction routines. If a particular surface is ON, the ceiling jet algorithm is used to determine the convective heating of the surface. If OFF, the bulk temperature of the upper layer determines the convective heating.

Key word: FAREA Inputs: Fuel Area	
Fuel Area (m)	The area of the fire at the base of the flames.

Key word: FHIGH Inputs: Fuel Height	
Fuel Height (m)	The height of the base of the flames above the floor of the compartment of fire origin for each point of the specified fire.

Key word: FMASS Inputs: Mass Loss Rate	
Mass Loss Rate (kg/s)	The rate at which fuel is pyrolyzed at times corresponding to each point of the specified fire.

Key word: FPOS Inputs: Depth, Breadth, Height (relative to the left rear corner of the compartment – see figure above)	
Depth	Position of the fire as a distance from the rear wall of the compartment (X direction).
Breadth	Position of the fire as a distance from the left wall of the compartment (Y direction).
Height	Height of the fire above the floor (Z direction). This value is simply added to the fire height at each time specified by the FHIGH key word.

Key word: FQDOT Inputs: Heat Release Rate	
Heat Release Rate (W)	The heat release rate of the specified fire.

Key word: FTIME Inputs: Time Points	
Time Points (s)	An entry indicates a point on the timeline where mass loss rate, fuel height and species are specified for the fire. This time is independent of the simulation time which is specified for the TIMES label. If the simulation time is longer than the total duration of the fire, the final values specified for the fire (mass loss rate, fuel height, fuel area, and species) are continued until the end of the simulation.

Key word: LFBO Input: Compartment of Fire Origin	
Compartment of Fire Origin	Compartment of fire origin is the compartment number in which the fire originates. Default is 0. The outside can not be specified as a compartment. An entry of 0 turns off the main fire leaving only object fires specified by the OBJECT key word.

Key word: LFBT Input: Fire Type	
Fire Type	This is a number indicating the type of fire. 1 Unconstrained fire 2 Constrained fire. The default is 1. See section 3.1 for a discussion of the implications of this choice.

Key words: HCN, HCL, CT, HCR, or O2 Inputs: Composition of the Pyrolyzed Fuel	
Production Rate (kg/kg)	Units are kilogram of species produced per kilogram of fuel pyrolyzed for HCN and HCL. The input for CT is the kilograms of "toxic" combustion products produced per kilogram of fuel pyrolyzed. Input for HCR is the mass ratio of hydrogen to carbon and oxygen to carbon *as it becomes available from the fuel* for O2.

Key words: OD and CO Inputs: Yield	
Yield (kg/kg)	Input the ratio of the mass of carbon to carbon dioxide produced by the oxidation of the fuel for OD. The input for CO is the ratio of the mass of carbon monoxide to carbon dioxide produced by the oxidation of the fuel.

6.2.9 Objects

The OBJECT key word allows the specification of additional objects to be burned in the fire scenario. The object name and object compartment are required if the OBJECT key word is used. All other input items have default values if they are not specified. These defaults are: start time 0.0, first element 1, depth (x position) one half the depth of the compartment, breadth (y position) one half the width of the compartment, and height (z position) 0.0. To specify any input item, all preceding items on the OBJECT line must also be specified. For example, the first element can not be set if start time is not set. Positioning of the object within a compartment is specified in the same manner as for the main fire (Figure 27).

EXAMPLE:

```
OBJECT SOFA       1     10   1   4.00   2.00   0.00
OBJECT WARDROBE   1     30   3   0.00   2.00   0.00
```

Key word: OBJFL Input: Objects Database	
Objects Database	The name specifies a file (up to 17 characters) from which the program obtains object data. If this parameter is not specified, then either the default (OBJECTS.DAT) is used, or the name is read from the configuration file.

Key word: OBJECT Inputs: Object Name, Object Compartment, Start Time, First Element, Depth, Breadth, Height	
Object Name	The name from the objects database for the desired object. Specifying a name not found in the database causes CFAST to stop with an appropriate error message. CEdit considers such an object undefined and does not display the entry.
Object Compartment	The compartment that the object is in during the simulation. If a compartment number outside the range of specified compartments is used, CFAST provides an error message and stops. CEdit considers such an object undefined and does not display the entry.
Start Time	The earliest time during the simulation when the CFAST model should check the fire scenario to determine if conditions have been met for the object to begin burning. If the flux to the object and the surface temperature are high enough, the object starts to burn.
First Element	Indicates the position on the object's surface where burning starts. This value is not currently used but needs to be set for future compatibility with the CFAST model. If a value is not known, enter a value of 1.
Depth	Position of the object as a distance from the rear wall of the object compartment (X direction).
Breadth	Position of the object as a distance from the left wall of the object compartment (Y direction)
Height	Height of the object above the floor (Z direction).

6.2.10 Files

There are several files which CFAST uses to communicate with its environment. They include 1) a configuration file, 2) the thermal database, 3) the objects database, 4) a history file, and 5) a restart file. The format of the thermal database and objects database are detailed in the User's Guide [14]. To access an alternate database see the appropriate input file section above.

The output of the simulation may be written to a disk file for further processing by programs such as CPlot or to restart CFAST. At each interval of time as specified by the history interval in the TIMES label, the output is written to the file specified. For efficient disk storage and optimum speed, the data is stored in an internal format and cannot be read directly with a text editor.

Example:

```
DUMPR PRM.HI
```

Key word: DUMPR Input: History File	
History File	The name specifies a file (up to 17 characters) to which the program outputs for plotting are written. History file is an optional input. If omitted, the file will not be generated. Note that in order to obtain a history of the variables, this parameter must be specified, and the history interval (see Section 6.2.1) must be set to a non-zero value.

Key word: RESTR Input: Restart File, Restart Time (see Section 6.2.1)	
Restart File	The name specifies a file (up to 17 characters) from which the program reads data to restart the model. This data must have been generated (written) previously with the history parameter discussed earlier.

6.3 Additional Key Words Unavailable in CEdit

6.3.1 Fire Specification

The LFPOS key word is provided to maintain backward compatibility with previous versions of FAST and CFAST. For exact positioning of the main fire, use the FPOS key word in section 6.2.8. The CHEMI line discussed in section 6.2.8 corresponds to chemical kinetics entry available in CEdit. At this time, CEdit does not support entry of the initial fuel temperature, gaseous ignition temperature, or the radiative fraction. These parameters must be entered on the CHEMI line using an ASCII text editor.

Example:

```
CHEMI  16.    0.  10.0    18100000.  300. 400.   0.
LFPOS  1
```

Key word: CHEMI Inputs: Molar Weight (see sec. 6.2.8), Relative Humidity (see sec. 6.2.8), Lower Oxygen Limit (see sec. 6.2.8), Heat of Combustion (See sec. 6.2.8), Initial Fuel Temperature, Gaseous Ignition Temperature, Radiative Fraction	
Initial Fuel Temperature (K)	Typically, the initial fuel temperature is the same as the ambient temperature as specified in the ambient conditions section.
Gaseous Ignition Temperature (K)	Minimum temperature for ignition of the fuel as it flows from a compartment through a vent into another compartment. If omitted, the default is arbitrarily set to the initial fuel temperature plus 100K.
Radiative Fraction	The fraction of heat released by the fire that goes into radiation. Default is 0.15.

Key word: LFPOS Input: Fire Position	
Fire Position	The fire position indicates where in the compartment of fire origin the fire originates and is one of the following values: 1 Center of the compartment, 2 Corner of the compartment, or 3 Along a wall of the compartment, but not near a corner of the compartment. The fire position is used to account for the entrainment rate of the plume, which depends on the location of the fire plume within the compartment. Fire positions 2 and 3 should only be used when the fire is very close to the corner or wall respectively. The default is 1.

6.3.2 Graphics Specification

A graphics specification can be added to the data file. Details of the meaning of some of the parameters is best left to the discussion of the device independent graphics software used by CFAST [77]. However, the information necessary to use it is straightforward. The general structure is similar to that used for the compartment and fire specification. One must tell the program "what to plot," "how it should appear," and "where to put it."

The key words for "where to put it" are:

DEVICE	where to plot it
BAR	bar charts
GRAPH	specify an x-y plot
TABLE	put the data into a table
PALETTE	specify the legend for CAD views
VIEW	show a perspective picture of the structure
WINDOW	the size of the window in "user" space.

The complete key word is required. That is, for the "where to put it" terms, no abbreviations are allowed. Then one must specify the variables to be plotted. They are:

VENT, HEAT, PRESSUR, WALL, TEMPERA, INTERFA,
H_2O, CO_2, CO, OD, O_2, TUHC, HCN, HCL, CT

As might be expected, these are similar key words to those used in the plotting program, CPlot. In this case, it is a reduced set. The application and use of CFAST and CPlot are different.

For each key word there are parameters to specify the location of the graph, the colors and finally, titles as appropriate. For the variables, there is a corresponding pointer to the graph of interest.

The WINDOW label specifies the user space for placement of graphs, views, etc. The most common values (which are also the default) are:

Xl = 0., Yb = 0., Zf = 0.
Xr = 1279., Yt = 1023., Zb = 10.

This is not a required parameter; however, it is often convenient to define graphs in terms of the units that are used. For example, if one wished to display a house in terms of a blueprint, the more natural units might be feet. In that case, the parameters might have the values:

Xl = 0., Yb = 0., Zf = 0.
Xr = 50., Yt = 25., Zb = 30.

Up to five graphs, tables, bar charts, and views may be displayed at one time on the graphics display. Up to five labels may be displayed at one time on the graphics display. Each type of output and each label is identified by a unique number (1-5) and placed in the window at a specified location. Xl, Yb, Zf, Xr, Yt and Zb have a meaning similar to WINDOW. However, here they specify where in the window to put the output.

The PALETTE label performs a specialized function for showing colors on the views. A four entry table is created and used for each type of filling polygon used in a view. Up to five palettes may be defined. Each palette is identified by a unique number and placed in the window at a specified location. Xl, Yb, Zf, Xr, Yt and Zb have a meaning similar to WINDOW. However, here they specify where in the window to put the palette.

In order to see the variables, they must be assigned to one of the above displays. This is accomplished with the variable pointers as:

(Variable) (nmopq) (Compartment) (Layer).

Variable is one of the available variables VENT, HEAT, PRESSUR, WALL, TEMPERA, INTERFA, N_2, O_2, CO_2, CO, HCN, HCL, TUHC, H_2O, OD, CT used as a label for the line. The species listed correspond to the variable "SPECIES" in CPlot. (nmopqr) is a vector which points to:

```
+)))))))))))))))))))),
* index   display in  *
/))))))))))))))))))))1
* (1) n -> bar chart  *
* (2) m -> table      *
* (3) o -> view       *
* (4) p -> label      *
* (5) q -> graph      *
.))))))))))))))))))))-
```

respectively. These numbers vary from 1 to 5 and correspond to the value of "n" in the "where to put it" specification. Compartment is the compartment number of the variable and Layer is "U" or "L" for upper and lower layer, respectively.

Example:

```
WINDOW       0     0  -100  1280  1024   1100
GRAPH 1  100.  050.  0.  600.  475.  10.  3 TIME  HEIGHT
GRAPH 2  100.  550.  0.  600.  940.  10.  3 TIME  CELSIUS
GRAPH 3  720.  050.  0.  1250. 475.  10.  3 TIME  FIRE_SIZE(kW)
GRAPH 4  720.  550.  0.  1250. 940.  10.  3 TIME  O|D2|O(%)
INTERFA 0 0 0 0 1    1 U
TEMPERA 0 0 0 0 2    1 U
HEAT    0 0 0 0 3    1 U
O2      0 0 0 0 4    1 U
INTERFA 0 0 0 0 1    2 U
TEMPERA 0 0 0 0 2    2 U
HEAT    0 0 0 0 3    2 U
O2      0 0 0 0 4    2 U
```

Key word: DEVICE Input: Plotting Device	
Plotting Device	The Plotting Device specifies the hardware device where the graphics is to be displayed. It is installation dependent. In general it specifies which device will receive the output. For most systems, 1 is for the screen from which keyboard input comes, and 6 is for the hpgl files.

Key word: WINDOW Inputs: Xl, Yb, Zf, Xr, Yt, Zb	
Xl	Left hand side of the window in any user desired units.
Yb	Bottom of the window in any user desired units.
Zf	Forward edge of the 3D block in any user desired units.
Xr	Right hand side of the window in any user desired units.
Yt	Top of the window in any user desired units.
Zb	Rear edge of the 3D block in any user desired units. These definitions refer to the 3D plotting block that can be seen.

Key word: BAR Inputs: Bar Chart Number, Xl, Yb, Zf, Xr, Yt, Zb, Abscissa Title, Ordinate Title	
Bar Chart Number	The number to identify the bar chart. Allowable values are from 1 to 5.
Xl	Left hand side of the bar chart within the window in the same units as that of the window.
Yb	Bottom of the bar chart within the window in the same units as that of the window.
Zf	Forward edge of the 3D block within the window in the same units as that of the window.
Xr	Right hand side of the bar chart within the window in the same units as that of the window.
Yt	Top of the bar chart within the window in the same units as that of the window.
Zb	Back edge of the 3D block within the window in the same units as that of the window.
Abscissa Title	Title for the abscissa (horizontal axis). To have blanks in the title, use the underscore character "_".
Ordinate Title	Title for the ordinate (vertical axis). To have blanks in the title, use the underscore character "_".

Key word: GRAPH Inputs: Graph Number, Xl, Yb, Zf, Xr, Yt, Zb, Color, Abscissa Title, Ordinate Title	
Graph Number	The number to identify the graph. Allowable values are from 1 to 5. The graphs must be numbered consecutively, although they do not have to be given in order. It is acceptable to define graph 4 before graph 2, but if graph 4 is to be used, then graphs 1 through 3 must also be defined.
Xl	Left hand side of the graph within the window in the same units as that of the window.
Yb	Bottom of the graph within the window in the same units as that of the window.
Zf	Forward edge of the 3D (three dimensional) block within the window in the same units as that of the window.
Xr	Right hand side of the graph within the window in the same units as that of the window.
Yt	Top of the graph within the window in the same units as that of the window.
Zf	Back edge of the 3D block within the window in the same units as that of the window.
Color	The color of the graph and labels which is specified as an integer from 1 to 15. Refer to DEVICE (NBSIR 85-3235) for the colors corresponding to the color values.
Abscissa Title	Title for the abscissa (horizontal axis). To have blanks in the title, use the underscore character "_".
Ordinate Title	Title for the ordinate (vertical axis). To have blanks in the title, use the underscore character "_".

Key word: TABLE Inputs: Table Number, Xl, Yb, Zf, Xr, Yt, Zb	
Table Number	The table number is the number to identify the table. Allowable values are from 1 to 5. The tables must be numbered consecutively, although they do not have to be given in order. It is acceptable to define table 4 before table 2, but if table 4 is to be used, then tables 1 through 3 must also be defined.
Xl	Left hand side of the table within the window in the same units as that of the window.
Yb	Bottom of the table within the window in the same units as that of the window.
Zf	Forward edge of the 3D block within the window in the same units as that of the window.
Xr	Right hand side of the table within the window in the same units as that of the window.
Yt	Top of the table within the window in the same units as that of the window.
Zb	Back edge of the 3D block within the window in the same units as that of the window.

Key word: VIEW Inputs: View Number, Xl, Yb, Zf, Xr, Yt, Zb, File, Transform Matrix	
View Number	View number is the number to identify the view. Allowable values are from 1 to 5. The views must be numbered consecutively, although they do not have to be given in order. It is acceptable to define view 4 before view 2, but if view 4 is to be used, then views 1 through 3 must also be defined.
Xl	Left hand side of the view within the window in the same units as that of the window.
Yb	Bottom of the view within the window in the same units as that of the window.
Zf	Forward edge of the 3D block within the window in the same units as that of the window.
Xr	Right hand side of the view within the window in the same units as that of the window.
Yt	Top of the view within the window in the same units as that of the window.
Zf	Back edge of the 3D block within the window in the same units as that of the window.
File	File is the filename of a building descriptor file.
Transform Matrix	The Transform Matrix is a 16 number matrix which allows dynamic positioning of the view within the window. The matrix (1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1) would show the image as it would appear in a display from BUILD.

Key word: LABEL Inputs: Label Number, Xl, Yb, Zf, Xr, Yt, Zb, Text, Angle1, Angle2	
Label Number	Label number is the number to identify the label. Allowable values are from 1 to 5.
Xl	Left hand side of the label within the window in the same units as that of the window.
Yb	Bottom of the label within the window in the same units as that of the window.
Zf	Forward edge of the 3D block within the window in the same units as that of the window.
Xr	Right hand side of the label within the window in the same units as that of the window.
Yt	Top of the label within the window in the same units as that of the window.
Zb	Back edge of the 3D block within the window in the same units as that of the window.
Text	The text to be displayed within the label. To have blanks in the title, use the underscore character "_".
Color	Color of the text to be displayed (a number from 0 to 15).
Angle1 and Angle2	Angles for display of the label in a right cylindrical coordinate space. At present, only the first angle is used and represents a positive counter-clockwise rotation; set the second angle to zero. Both angles are in radians.

Key word: PALETTE Inputs: Palette Number, Xl, Yb, Zf, Xr, Yt, Zb, Color and Label	
Palette Number	Palette number is the number to identify the palette. Allowable values are from 1 to 5.
Xl	Left hand side of the palette within the window in the same units as that of the window.
Yb	Bottom of the palette within the window in the same units as that of the window.
Zf	Forward edge of the 3D block within the window in the same units as that of the window.
Xr	Right hand side of the palette within the window in the same units as that of the window.
Yt	Top of the palette within the window in the same units as that of the window.
Zb	Back edge of the 3D block within the window in the same units as that of the window.
Color and Label	There are four pairs of color/text combinations, each corresponding to an entry in the palette. The color number is an integer from 1 to 15 and the text can be up to 50 characters (total line length of 128 characters maximum). As before, spaces are indicated with an underscore character "_".

7 Conclusions and Future Plans

CFAST is a zone model used to calculate the evolving distribution of smoke and fire gases and the temperature throughout a building during a fire. Although it may not be all inclusive, CFAST has demonstrated the ability to make reasonably good predictions. Also, it has been subject to close scrutiny to insure its correctness. Thus it forms a paradigm for what constitutes a reasonable approach to modeling fire growth and the spread of smoke and toxic gases.

With this level of detail, researchers not intimately involved in the development of CFAST should be able to add to the model in a straightforward manner. Independent or cooperative efforts to enhance the capabilites of the model are encouraged. Model developers can use this version of the model for open or proprietary additions to the model or as the basis for new models. While encouraging additions to the model, the role of the National Institute of Standards and Technology (NIST) in the development must be clearly defined. NIST will continue to develop and document algorithms and the model in total. Depending on the extent others validate and document their additions to the model in the refereed literature, NIST may include them in its future versions. Thus, non-developers will continue to have access to an increasingly capable model from NIST, and perhaps from others as well.

The quest is to develop a tool which will help improve the understanding of fires. This is not an attempt to make the application of models trivial, but rather to provide a mechanism to allow researchers, fire protection engineers, and others access to the most current understanding of the behavior of fires. With this version of the model, we hope to provide a framework for cooperative development of the model by BFRL and outside researchers. To reach this goal, we outline possible directions for the future development of the model. This discussion is not intended to limit model development to these directions, but rather to stimulate thought in the possible improvements.

7.1 Capabilities and Processing Power

For fire investigation, we could have a portable computer (hand held) which allowed one to walk through a building (before or after) and catalog the contents of a building. This could be brought back to the office and used directly as input to the model for geometric specification and data initialization. As the model becomes more sophisticated, and the complexity increases, researchers, code officials, and others may need to depend on such stratagems. There simply is not enough time to fuss with all of the details. This is the arena which should allow us to pursue the goal of a better qualitative understanding of fires, and well as doing more of it faster.

All large buildings have breakout panels for various alarms. Indeed, some fire departments can display floor plans of buildings in the command center at a fire. It is a logical step to plug these building detections and alarm systems into CFAST for obtaining the current status and predictions of the building environment as the fire develops.

Another area is that of risk. Risk is the next step up from a hazard calculation, and requires a much more general understanding of the parameters which affect the outcome of a fire and its impact on humans and structures. This application would require an automated application of the model over types of fires, day and night scenarios, position of the fire and so on. The number of such calculations can become enormous. Some means of doing this in finite time will need to be found.

7.2 Model Improvements

This deals with improvements in the model itself. Beyond what we have today, we see the following as minimum improvements to present the concept to the whole of the fire related community. (We include in this general audience the code officials, FPE's, and designers.):

Construction design files (databases used for building and ship design)
Self consistent fire - both a flame spread model and a pyrolysis model
Species generation CO/CO_2, that considers vitiation effects on combustion and toxicity
General radiation model - odd shapes and heat transfer between compartments
Two directional heat transfer in walls (non-congruent thermocline)
Better detector and other sensor activation (include new detectors)
Agglomeration and deposition of smoke
Suppression - include fire size, drop size and distance effects, geometry of the fire (hidden)
Multiple layers and zones (hybrid modification) - a must for detector siting
Experimental correlations for flow up shafts and *stairways*
Modifications to all modules to utilize databases
Corrosion - add on for HCl - important for semiconductor industry and warehouses
Simple (and quick) estimates

The accuracy of the current procedure is limited by the fire being uninfluenced by radiation from its surroundings, and by our inability to quantify accurately the effects of fire on people and their actions. Research is underway to better understand radiation enhanced burning under post flashover conditions, and predict fire growth and spread, fuel mass loss rate and combustion product generation rates under those conditions. In general, the CFAST model provides reasonable predictions of the several experiments examined in this paper. Although differences between the model and the experiments are clear, they can be explained by limitations of the model and of the experiments. Several areas which need additional research are apparent:

- *Entrainment* – fire plume and doorway jet entrainment are based on the same experimental correlations. The fire plume (for large spaces) and the doorway jet (in general) are often used outside the normal range of validity of these correlations.

- *User specified fire* – the level of agreement is critically dependent upon careful choice of the input data for the model. A validated fire growth model would allow prediction of pyrolysis rate and species yields consistent with changing conditions during the fire. A better understanding of typical fire induced leakage in buildings would facilitate more accurate description of the building environment.

- *Statistical treatment of the data* – presentation of the differences between model predictions and experimental data in section 5 are intentionally simple. With a significant base of data to study, appropriate statistical techniques to provide a true measure of the “goodness of fit” should be investigated.

- *Experimental measurements* – measurement of leakage rates, room pressure, or profiles of gas concentration are atypical in experimental data. These measurements are critical to assessing the

accuracy of the underlying physics of the models and of the models ability to predict toxic gas hazard.

As with any theoretical model, there are pieces which have been omitted and others which could be implemented more completely. However, with an understanding of the relative weaknesses and strengths of both the model and of the experiments used to verify the model, the user can develop confidence in using such models for a wide range of simulations.

8 References

1. Friedman, R., "Survey of Computer Models for Fire and Smoke," *Factory Mutual Research Corp.*, Norwood, MA, 02062 1990.

2. Cooper, L.Y., "A Mathematical Model for Estimating Available Safe Egress Time in Fires," *Fire and Materials.* 1982, *6(4),* 135-144.

3. Babrauskas, V., "COMPF2-A Program for Calculating Post-Flashover Fire Temperatures," *Natl. Bur. Stand. (U.S.)* 1979, *Tech. Note 991*, 76 p.

4. Davis, W. D. and Cooper, L. Y., "Computer Model for Estimating the Response of Sprinkler Links to Compartment Fires With Draft Curtains and Fusible Link-Actuated Ceiling Vents," *Fire Technology* 1991, *27 (2)*, 113-127.

5. Mitler, H.E. and Emmons, H.W., "Documentation for CFCV, the Fifth Harvard Computer Fire Code," *Nat. Bur. Stand. (U.S.)* 1981, *NBSGCR 81-344*, 187 p.

6. Mitler, H. E. and Rockett, J., "A User's Guide for FIRST, a Comprehensive Single Room Fire Model," *Natl. Bur. Stand. (U. S.)* 1987, *NBSIR 87-3595*.

7. Tanaka, T., "A Model of Multiroom Fire Spread," *Nat. Bur. Stand. (U.S.)* 1983, *NBSIR 83-2718*, 175 p.

8. Gahm, J. B., "Computer Fire Code VI, Volume I," *Nat. Bur. Stand. (U.S.)* 1983, *NBS GCR 83-451*, 116 p.

9. Jones, W. W., A Multicompartment Model for the Spread of Fire, Smoke and Toxic Gases, Fire Safety Journal *9*, 55 (1985); Jones, W. W. and Peacock, R. D., Refinement and Experimental Verification of a Model for Fire Growth and Smoke Transport, Proceedings of the 2nd International Symposium on Fire Safety Science, Tokyo (1989); Jones, W. W. and Peacock, R. D., "Technical Reference Guide for FAST Version 18" Natl. Inst. Stand. Techol. Tech. Note 1262 (1989).

10. Forney, G. P. and Cooper, L. Y., The Consolidated Compartment Fire Model (CCFM) Computer Application CCFM.VENTS - Part II: Software Reference Guide, Nat. Inst. Stand. Technol., NISTIR 90-4343 (1990).

11. Jones, W. W. and Forney, G. P. "A Programmer's Reference Manual for CFAST, the Unified Model of Fire Growth and Smoke Transport," *Natl. Inst. Stand. Technol.* 1990, *Tech. Note 1283*, 104 p.

12. Mitler, H. E. "Comparison of Several Compartment Fire Models: An Interim Report," *Natl. Bur. Stand. (U.S.)* 1985, *NBSIR 85-3233*, 33 p.

13. Jones, W. W. "A Review of Compartment Fire Models," *Natl. Bur. Stand. (U.S.)* 1983, *NBSIR 83-2684*, 41 p.

14. Portier, R., Reneke, P. A., Jones, W. W., and Peacock, R. D., A User's Guide for CFAST, Version 1.6. Natl. Inst. Stand. Technol., NISTIR 4985 (1992).

15. Babrauskas, V., Lawson, J. R., Walton, W. D., Twilley, W. H., "Upholstered Furniture Heat Release Rates Measured with a Furniture Calorimeter," Natl. Bur. Stand. (U.S.), NBSIR 82-2604, 1982.

16. Babrauskas, V. and Krasny, J. F., "Fire Behavior of Upholstered Furniture," Natl. Bur. Stand. (U.S.), Monogr. 173, 1985.

17. Lee, B.T., "Effect of Ventilation on the Rates of Heat, Smoke, and Carbon Monoxide Production in a Typical Jail Cell Fire," Natl. Bur. Stand. (U.S.), NBSIR 82-2469, 1982.

18. Peacock, R. D., Davis, S., Lee, B. T., "An Experimental Data Set for the Accuracy Assessment of Room Fire Models," Natl. Bur. Stand. (U.S.), NBSIR 88-3752, April 1988, p. 120.

19. Cooper L. Y. and Forney, G. P., The consolidated compartment fire model (CCFM) computer application CCFM-VENTS – part I: Physical reference guide. Natl. Inst. Stand. Technol., NISTIR 4342 (1990).

20. Forney, G. P. and Moss, W. F., Analyzing and Exploiting Numerical Characteristics of Zone Fire Models, Natl. Inst. Stand. Technol., NISTIR 4763 (March 1992).

21. Jones, W. W. and Bodart, X., Buoyancy Driven Flow as the Forcing Function of Smoke Transport Models, Natl. Bur. Stand. (U. S.), NBSIR 86-3329 (1986).

22. Rehm, R. G. and Forney, G. P., “A Note on the Pressure Equations Used in Zone Fire Modeling,” Natl. Inst. Stand. Technol., NISTIR 4906 (1992).

23. Babrauskas, V., "Development of the Cone Calorimeter - A Bench Scale Heat Release Rate Apparatus Based on Oxygen Consumption," *Fire and Materials 8,* 1984, p 81.

24. "Standard Test Method for Heat and Visible Smoke Release for Materials and Products Using and Oxygen Consumption Calorimeter," ASTM E1354-90, *American Society for Testing and Materials,* Philadelphia, PA 1990.

25. Morehart, J. H., Zukowski, E. E., and Kubota, T., "Characteristics of Large Diffusion Flames Burning in a Vitiated Atmosphere," Third International Symposium on Fire Safety Science, Edinburgh 1991.

26. McCaffrey, B. J., "Momentum Implications for Buoyant Diffusion Flames," *Combustion and Flame 52,* 1983, p. 149.

27. Cetegen, B. M., "Entrainment and Flame Geometry of Fire Plumes," Ph.D. Thesis, California Institute of Technology, Pasadena 1982.

28. Tewarson, A., Combustion of Methanol in a Horizontal Pool Configuraiton, Factory Mutual Research Corp., Norwood, MA, Report No. RC78-TP-55 (1978).

29. McCaffrey, B. J., Entrainment and Heat Flux of Bouyant Diffusion Flames, Natl. Bur. Stand. (U. S.), NBSIR 82-2473, 35 p. (1982).

30. Koseki, H., Combustion Properties of Large Liquid Pool Fires, Fire Technology, **25(3)**, 241-255 (1989).

31. Quintiere, J. G., Steckler, K., and Corley, D., An Assessment of Fire Induced Flows in Compartments, Fire Science and Technology *4*, 1 (1984).

32. Quintiere, J. G., Steckler, K. and McCaffrey, B. J., "A Model to Predict the Conditions in a Room Subject to Crib Fires," First Specialist Meeting (International) of the Combustion Institute, Talence, France 1981.

33. Cooper, L. Y., Calculation of the Flow Through a Horizontal Ceiling/Floor Vent, Natl. Inst. Stand. Technol., NISTIR 89-4052 (1989).

34. Klote, J. H., A Computer Model of Smoke Movement by Air Conditioning Systems, NBSIR 87-3657 (1987).

35. 1989 ASHRAE Handbook Fundamentals, American Society of Heating, Refrigeration and Air Condition Engineers, Inc., Atlanta, GA 1989.

36. Forney, G. P., Computing Radiative Heat Transfer Occurring in a Zone Fire Model, Natl. Inst. Stand. Technol., NISTIR 4709 (1991).

37. Siegel, R. and Howell, J. R., Thermal Radiation Heat Transfer, Hemisphere Publishing Corporation, New York, second ed. (1981).

38. Hottel, H. C., Heat Transmission, McGraw-Hill Book Company, New York, third ed. (1954).

39. Hottel, H. and Cohen, E., Radiant Heat Exchange in a Gas Filled Enclosure: Allowance for non-uniformity of Gas Temperature, American Institute of Chemical Engineering Journal *4*, 3 (1958).

40. Yamada, T. and Cooper, L. Y., Algorithms for Calculating Radiative Heat Exchange Between the Surfaces of an Enclosure, the Smoke Layers and a Fire, Building and Fire Research Laboratory Research Colloquium, July 20, 1990.

41. Schlichting, H., "Boundary Layer Theory," translated by J. Kestin, Pergammon Press, New York, 1955.

42. Golub, G. H. and Ortega, J. M., Scientific Computing and Differential Equations, An Introduction to Numerical Methods. Academic Press, New York (1989).

43. Strang, G. and Fix, G. J., An Analysis of the Finite Element Method. Prentice-Hall, Englewood Cliffs, New Jersey (1973).

44. Moss, W. F. and Forney, G. P., “Implicitly coupling heat conduction into a zone fire model.” Natl. Inst. Stand. Technol., NISTIR 4886 (1992).

45. Cooper, L. Y., Fire-Plume-Generated Ceiling Jet Characteristics and Convective Heat Transfer to Ceiling and Wall Surfaces in a Two-Layer Zone-Type Fire Environment, Natl. Inst. Stand. Technol., NISTIR 4705, 57 p. (1991).

46. Cooper, L. Y., Heat Transfer in Compartment Fires Near Regions of Ceiling-Jet Impingement on a Wall. J. Heat Trans., 111, pp. 455-460 (1990).

47. Cooper, L. Y., Ceiling Jet-Driven Wall Flows in Compartment Fires. Combustion Sci. and Technol., 62, pp. 285-296 (1988).

48. Jaluria, Y. and Cooper, L. Y., Negatively Buoyant Wall Flows Generated in Enclosure Fires. Progress in Energy and Combustion Science, 15, pp. 159-182 (1989).

49. Galloway, F. M., Hirschler, M. M., "A Model for the Spontaneous Removal of Airborne Hydrogen Chloride by Common Surfaces," *Fire Safety Journal 14*, 1989, pp. 251-268.

50. Galloway, F. M., Hirschler, M. M., "Transport and Decay of Hydrogen Chloride: Use of a Model to Predict Hydrogen Chloride Concentrations in Fires Involving a Room-Corridor-Room Arrangement," *Fire Safety Journal 16*, 1990, pp. 33-52.

51. Brenan, K. E., Campbell, S. L., and Petzold, L. R., Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations, Elsevier Science Publishing, New York (1989).

52. Alpert, R. L., et al., "Influence of Enclosures on Fire Growth: Volume I: Test Data," *Factory Mutual Research Corp.,* Norwood MA. 1977, *FMRC No. OAOR2.BU-1 through 7.*

53. Cooper, L. Y., Harkleroad, M., Quintiere, J. G., and Rinkinen, W. J., "An Experimental Study of Upper Hot Layer Stratification in Full-Scale Multiroom Fire Scenarios," *J. Heat Trans.* 1982, *104*, 741-749.

54. Quintiere, J. G., and McCaffrey, B. J., "The Burning of Wood and Plastic Cribs in an Enclosure," 2 vols., *Nat. Bur. Stand. (U.S.)* 1980, *NBSIR 80-2054*.

55. Heskestad, G., and Hill, J. P., "Experimental Fires in Multiroom / Corridor Enclosures," *Natl. Bur. Stand. (U.S.)* 1986, *NBSGCR 86-502, CIB W14/85/10 (USA).*

56. Rockett, J. A., Morita, M., and Cooper, L. Y., "Comparisons of NBS/Harvard VI Simulations and Full-scale Multi-room Fire Test Data," in *Proceedings of 2nd International Symposium of Fire Safety Science*, Tokyo, Japan. 1988, 481-490.

57. Rockett, J. A., Morita, M., and Cooper, L. Y., "Comparisons of NBS/Harvard VI Simulations and Data from all Runs of a Full-scale Multi-room Fire Test Program," *Fire Safety J.* 1989, *15*, 115-169.

58. Rockett, J. A., and Morita, M., "The NBS Harvard VI Multi-room Fire Simulation," *Fire Sci. Technol.* 1985, *5(2)*, 159-164.

59. Peacock, R. D., Davis, S., and Babrauskas, V., Data for Room Fire Model Comparisons. J. Res. Natl. Inst. Stand. Technol. **96**, 411 (1991).

60. Nelson, H. E., "An Engineering View of the Fire of May 4, 1988 in the First Interstate Bank Building, Los Angeles, California," *Natl. Inst. Stand. Technol.* 1989, *NISTIR 89-4061*, 39 p.

61. Bukowski, R. W., "Reconstruction of a Fatal Residential Fire at Ft. Hood, Texas," *First HAZARD I Users' Conference*, Natl. Inst. Stand. Technol., Gaithersburg, MD, June 5-6, 1990.

62. Emmons, H.W., "Why Fire Model? The MGM Fire and Toxicity Testing," *Fire Safety J.*, **13** 77-85 1988.

63. Levine, R. S. and Nelson, H. E., "Full Scale Simulation of a Fatal Fire and Comparison of Results with Two Multiroom Models," *Natl. Inst. Stand. Technol.* 1990, *NISTIR 90-4268*, 105 p.

64. Deal, S. "A Review of Four Compartment Fires with Four Compartment Fire Models," *Fire Safety Developments and Testing*, Proceedings of the Annual Meeting of the Fire Retardant Chemicals Association. October 21-24, 1990, *Ponte Verde Beach, Florida*, 33-51.

65. Nelson, H. E., "FPETOOL: Fire Protection Engineering Tools for Hazard Estimation," *Natl. Inst. Stand. Technol.* 1990, *NISTIR 4380*, 120 p.

66. Duong, D. Q., "The Accuracy of Computer Fire Models: Some Comparisons with Experimental Data from Australia," *Fire Safety J.* 1990, *16(6)*, 415-431.

67. Babrauskas, V., "Upholstered Furniture Room Fires -- Measurements, Comparison with Furniture Calorimeter Data, and Flashover Predictions," *J. Fire Sci.* 1984, *4*, 5-19.

68. Lee, B. T., Effect of Wall and Room Surfaces on the Rates of Heat, Smoke, and Carbon Monoxide Production in a Park Lodging Bedroom Fire, *Natl. Bur. Stand.* (U. S.), 1985, *NBSIR 85-2998*, 78 p.

69. Heskestad, G., and Hill, J. P., "Propagation of Fire and Smoke in a Corridor," *Proceedings of the 1987 ASME:JSME Thermal Engineering Joint Conference* 1987, *Honolulu, HI*, 371-379.

70. Klote, J. H., "Fire Experiments of Zoned Smoke Control at the Plaza Hotel in Washington DC," *Natl. Inst. Stand. Technol.* 1990, *NISTIR 90-4253*, 75 p.

71. Babrauskas, V. B. and Peacock, R. D., "Heat Release Rate: The Single Most Important Variable in Fire Hazard." *Fire Safety J.* **18** (1992) 255-272.

72. Quintiere, J. G., and McCaffrey, B. J., "The Burning of Wood and Plastic Cribs in an Enclosure," *Natl. Bur. Stand. (U. S.)* 1980, *NBSIR 80-2954*, 2 vols.

73. Delichatsios, M. A., "Fire Growth Rate in Wood Cribs," *Combustion and Flame* 1976, *27*, 267-278.

74. Emmons, H. W., "Vent Flows," Section 1, Chapter 8 in The SFPE Handbook of Fire Protection Engineering, C. L. Beyler, Ed., *National Fire Protection Association*, Quincy, MA 1988.

75. U.S. Standard Atmosphere (1976), a joint publication of the National Oceanographic and Atmospheric Administration and the National Aeronautics and Space Administration.

76. Haltiner, G. J. and Martin, F. L., Dynamical and Physical Meteorology, York (1957).

77. Jones, W. W. and Fadell, A. B., "A Device Independent Graphics Kernel," Natl. Bur. Stand. (U. S.), NBSIR 85-3235 (1988)

Appendix A: CFAST Call Trees

Routines within each tree are displayed in call order based on the first occurrence of the routine. Each routine is displayed once even though repeated calls to the subroutine may occur in the parent routine. Alternate subtrees in the right column represent the call tree for the Unix platform.

NOTATION: `file1>file2` indicates an entry point `file1` within the source routine `file2`
{} indicates a library routine
@ indicates an external function

LIBRARIES: IBMPC - DEVICEEM,UTIL3,F77L3; Unix - DEVICEX,X11,BLAS,M

A.1 Main trees

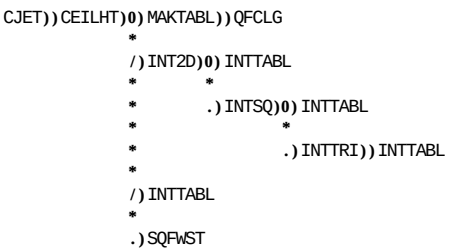
```
CFAST)0)INITMM
      *
      /)READOP (See Utility Subtree Section A.8)
      *
      /)OPENSHEL (See Utility Subtree Section A.8)
      *
      /)DISCLAIM
      *
      /)NPUTP (See Input Subtree Section A.6)
      *
      /)INITSPEC (See Initialization Subtree Section A.5)
      *
      /)NPUTT (See Input Subtree Section A.6)
      *
      /)INITSLV
      *
      /)INITWALL))WSET
      *
      /)NPUTOB (See Input Subtree Section A.6)
      *
      /)RESTRT (See Input Subtree Section A.6)
      *
      /)NPUTO (See Output Subtree Section A.7)
      *
      /)DISPLAYC (See Graphics Subtree Section A.4)
      *
      /)DUMPER (See Output Subtree Section A.7)
      *
      /)CPTIME))){ETIME}
      *
      /)SOLVE (See Numerical Subtree Section A.3)
      *
      .)ENDDIS>DISPLAYC)0){FRAME}
                         *
                         .){ENDFRM}
```

```
CPLOT)0)READOP (See Utility Subtree Section A.8)
      *
      /)OPENSHEL (See Utility Subtree Section A.8)
      *
      /){HSETSP}
      *
      /)INIPAR))F_RCOUNT>F_INFO
      *
      /){SYSTEM}
      *
      /)DISCLAIM
      *
      /)MESSNR>CSPACE)){PRINTS}
      *
      /)READASTU
      *
      /)SSTRNGP
      *
      /)READHI (See Input Subtree Section A.6)
      *
      /)MESS
      *
      /)OFFSET
      *
      /)DATACOPY
      *
      /)F_INFO))F_INFO16
      *
      /)LSTINFO))MESS
      *
      /)WHICHONE (See Graphics Subtree Section A.4)
      *
      /)R_TIME>F_INFO))FINFOR16>F_INFO16
      *
      /)OUTPU))OUTPU16
      *
      /)DORAPID (See Input Subtree Section A.6)
      *
      /)RDCNL
      *
      /)DOTENAB (See Input Subtree Section A.6)
      *
      .)RDTENA
```

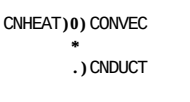
```
CEDIT)0)READOP (See Utility Subtree Section A.8)
      *
      /)OPENSHEL (See Utility Subtree Section A.8)
      *
      /){TIME}
      *
      /)DISCLAIM
      *
      /)IMOUSE (See Device Subtree Section A.10)
      *
      /)SET_UNT>SETUNT
      *
      /)CSPACE (See Device Subtree Section A.10)
      *
      /)CURSOF>CSPACE))){CUROFF}
      *
      /)NEWFILE (See Utility Subtree Section A.8)
      *
      /)INITFS (See Initialization Subtree Section A.5)
      *
      /)DISGEN (See Display Subtree Section A.9)
      *
      /)DISAMB (See Display Subtree Section A.9)
      *
      /)DISGEO (See Display Subtree Section A.9)
      *
      /)DISVT1 (See Display Subtree Section A.9)
      *
      /)DISVT3 (See Display Subtree Section A.9)
      *
      /)DISVT2 (See Display Subtree Section A.9)
      *
      /)DISTH1 (See Display Subtree Section A.9)
      *
      /)DISFIR (See Display Subtree Section A.9)
      *
      /)DISOB1 (See Display Subtree Section A.9)
      *
      /)DISCAL (See Display Subtree Section A.9)
      *
      /)DISPRM (See Display Subtree Section A.9)
      *
      /)MESSNS (See Utility Subtree Section A.8)
      *
      /)CHRMOV>CSPACE))){LOCATE}
      *
      .)MESSNR>CSPACE))){PRINTS}



REPORT)0)READOP (See Utility Subtree Section A.8)
       *
       /)OPENSHEL (See Utility Subtree Section A.8)
       *
       /)DISCLAIM
       *
       /)SSTRNG
       *
       /)READHI (See Input Subtree Section A.6)
       *
       /)INIPAR))F_RCOUNT>F_INFO
       *
       /)MESS
       *
       /)OFFSET
       *
       /)DATACOPY
       *
       /)NPUTOR (See Output Subtree Section A.7)
       *
       /)RSLT))FLWOUT
       *
       .)NAILED (See Physical Phenomenon Subtree Section A.2)
```

```
REPORTG)0)INITMM
        *
        /)READOP (See Utility Subtree Section A.8)
        *
        /)OPENSHEL (See Utility Subtree Section A.8)
        *
        /)DISCLAIM
        *
        /)NPUTP (See Input Subtree Section A.6)
        *
        /)INITSPEC (See Initialization Subtree Section A.5)
        *
        /)SSTRNG
        *
        /)READHI (See Input Subtree Section A.6)
        *
        /)INIPAR))F_RCOUNT>DISPLAYC
        *
        /)MESS
        *
        /)DISPLAYC (See Graphics Subtree Section A.4)
        *
        /)IOWAIT>CSPACE))<WAIT>
        *
        .)ENDDIS>DISPLAYC)0))<FRAME>
                          *
                          .)<ENDFRM>
```

A.2 Physical Phenomenon Subtrees



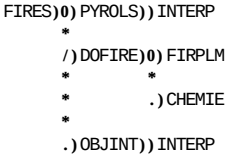
CALLED BY: RESID



CALLED BY: RESID

CALLED BY: RESID

CALLED BY: RESID



CALLED BY: RESID

CALLED BY: RESID

```
HFLOW)0)GETVAR
      *
      /)INTERP
      *
      /)VENT)0)GETELEV
      *      *
      *      .)DELP
      *
      /)FLOGO1
      *
      .)ENTRAIN))ENTRFL
```

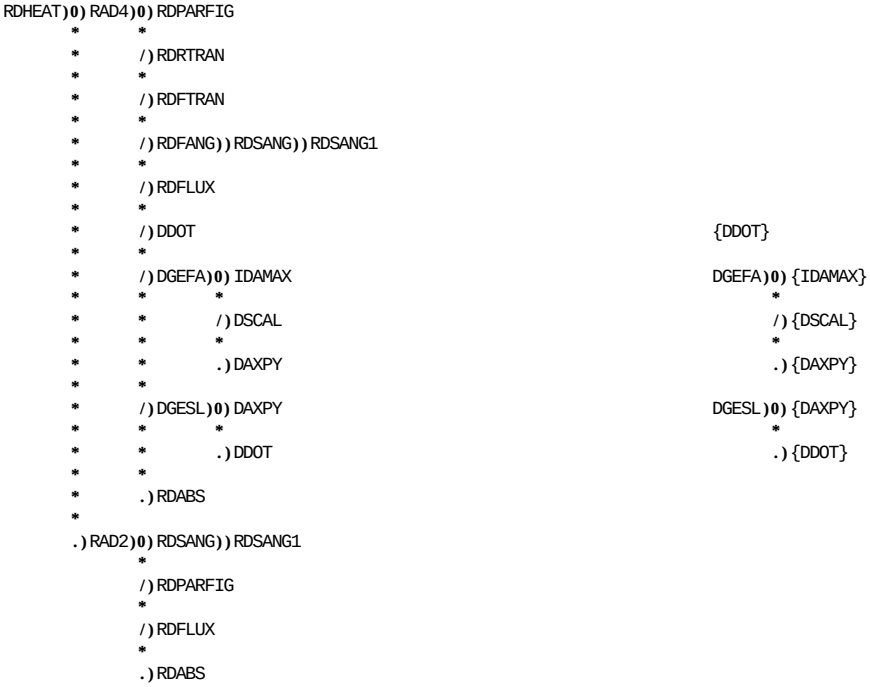
CALLED BY: RESID

```
MVENT)0)HVFREX
      *
      /)HVMFLO)0)HVVIS
      *         *
      *         /)HVFRIC
      *         *
      *         .)HVFAN
      *
      /)HVSFLO
      *
      .)HVTOEX
```

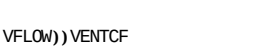
CALLED BY: RESID

```
NAILED)0)GETN))F_INFO))F_INFO16
       *
       /)INIPAR))F_RCOUNT>F_INFO
       *
       /)MATCH))FIND
       *
       /)CONT1)0)R_TIME>F_INFO))FINFOR16>F_INFO16
       *        *
       *        /)COPY
       *        *
       *        .)OUTPU1
       *
       /)DOSES)0)LAYERS))AVG
       *        *
       *        /)AVG
       *        *
       *        /)XFDCO
       *        *
       *        /)XFDO2
       *        *
       *        /)XFDHCN
       *        *
       *        /)FDCO
       *        *
       *        /)FDHCN
       *        *
       *        /)VDCO2
       *        *
       *        /)FDO2
       *        *
       *        /)CONHT
       *        *
       *        .)FDCO2
       *
       .)MESS
```

CALLED BY: REPORT

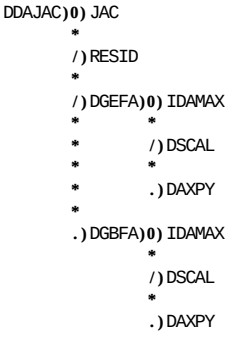


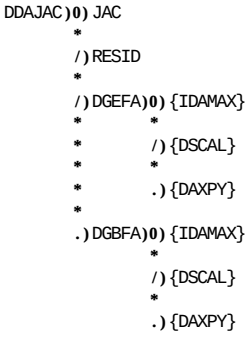
CALLED BY: RESID



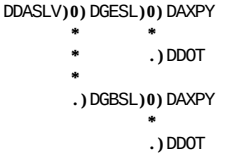
CALLED BY: RESID

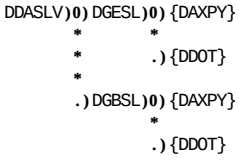
A.3 Numerical Subtrees



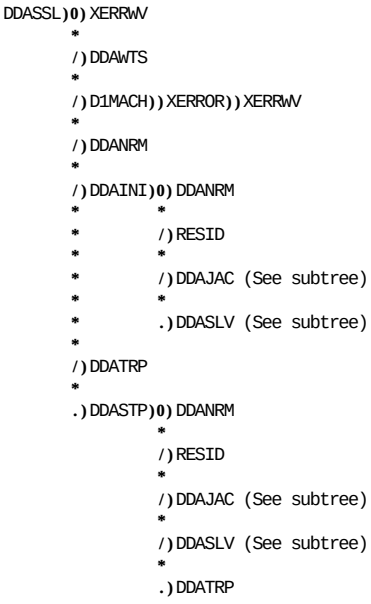


CALLED BY: DDAINI, DDASTP





CALLED BY: DDAINI, DDASTP



CALLED BY: SOLVE

```
INITSOLN)0)SNSQE)0)SNSQ)0)D1MACH))XERROR))XERRWV
         *       *      *
         *       *      /)@FCN (GRES or GRES2)
         *       *      *
         *       *      /)ENORM
         *       *      *
         *       *      /)GJAC
         *       *      *
         *       *      /)FDJAC1)0)D1MACH))XERROR))XERRWV
         *       *      *         *
         *       *      *         .)@FCN (GRES or GRES2)
         *       *      *
         *       *      /)QRFAC)0)D1MACH))XERROR))XERRWV
         *       *      *        *
         *       *      *        .)ENORM
         *       *      *
         *       *      /)QFORM
         *       *      *
         *       *      /)DOGLEG)0)D1MACH))XERROR))XERRWV
         *       *      *         *
         *       *      *         .)ENORM
         *       *      *
         *       *      /)R1UPDT))D1MACH))XERROR))XERRWV
         *       *      *
         *       *      /)R1MPYQ
         *       *      *
         *       *      .)XERROR))XERRWV
         *       *
         *       .)XERROR))XERRWV
         *
         .)RESID (See subtree)
```

CALLED BY: SOLVE

```
RESID)0)DATACOPY
      *
      /)HFLOW (See Physical Phenomenon Subtree Section A.2)
      *
      /)VFLOW (See Physical Phenomenon Subtree Section A.2)
      *
      /)MVENT (See Physical Phenomenon Subtree Section A.2)
      *
      /)FIRES (See Physical Phenomenon Subtree Section A.2)
      *
      /)SORTFR))INDEXI
      *
      /)DJET (See Physical Phenomenon Subtree Section A.2)
      *
      /)CJET (See Physical Phenomenon Subtree Section A.2)
      *
      /)CVHEAT (See Physical Phenomenon Subtree Section A.2)
      *
      /)RDHEAT (See Physical Phenomenon Subtree Section A.2)
      *
      /)HCL (See Physical Phenomenon Subtree Section A.2)
      *
      .)CNHEAT (See Physical Phenomenon Subtree Section A.2)
```

CALLED BY: GRES, GRES2, INITSOLN, SOLVE

```
SOLVE)0)INITSOLN (See subtree)
      *
      /)NTRACT)0)GRABKY>CSPACE (See Device Subtree Section A.10)              NTRACT)0){GRABKY}
      *         *                                                                    *
      *         /)SLVHELP))GRABKY>CSPACE (See Device Subtree Section A.10)         /)SLVHELP))GRABKY}
      *         *                                                                    *
      *         .)DEBUGPR))GRABKY>CSPACE (See Device Subtree Section A.10)         .)DEBUGPR))GRABKY}
      *
      /)RESULT))FLWOUT
      *
      /)DUMPER (See Output Subtree Section A.7)
      *
      /)DISPLAYC (See Graphics Subtree section A.4)
      *
      /)DEBUGPR))GRABKY>CSPACE (See Device Subtree Section A.10)              DEBUGPR))GRABKY}
      *
      /)CPTIME))TIMER}                                                       CPTIME))ETIME}
      *
      /)DDASSL (See subtree)
      *
      /)RESID (See subtree)
      *
      /)RESYNC
      *
      .)TOXIC
```

CALLED BY: CFAST

A.4 Graphics Subtrees

```
DISPLAYC)0)INIPAR))F_RCOUNT>DISPLAYC
         *
         /)MOUSE_OFF}
         *
         /)SETDVC}
         *
         /)HSETSP}
         *
         /)DEVICE}
         *
         /)NEWFRM}
         *
         /)COLOR}
         *
         /)DEF3D}
         *
         /)LINWID}
         *
         /)FILTYP}
         *
         /)GETVIEW)0)PLYPLT}
         *          *
         *          .)PLYGNS}
         *
         /)VWPRT
         *                                                                   * UNIX only
         /)*See subtree to right                                            /)XCHECK}
         *                                                                  *
         /)GRILAB>GRAFIT                                                    /)ERASE}
         *
         /)GRAFIT)0)BOXPLT}
         *         *
         *         /)WDCOUNT}
         *         *
         *         /)LABEL}
         *         *
         *         /)FNUMBR}
         *         *
         *         .)LINE}
         *
```

```
/)(BOXPLT}
*
/)(LABEL}
*
/)PALETTE)0)(FILTYP}
*          *
*          /)(COLOR}
*          *
*          /)(PLYGON}
*          *
*          .)(LABEL}
*
/)TOXICH)0)(COLOR}
*          *
*          /)(LINE}
*          *
*          /)CHRSET>WDDRAW
*          *
*          .)(LABEL}
*
/)TOXICB)0)(COLOR}
*          *
*          .)(LABEL}
*
/)OUTPU1
*
/)AHISTS
*
/)BXBLIT)0)(FILTYP}
*          *
*          /)(COLOR}
*          *
*          .)(PLYGON}
*
/)(LINTYP}
*
/)PLOTLN>GRAFIT))(LNPLOT}
*
/)TOXICR)0)(FILTYP}
*          *
*          /)(COLOR}
*          *
*          /)(PLYGON}
*          *
*          .)(LABEL}
*
/)(VIEWTR}
*
/)(FRAME}
*
/)(HDCOPY}
*
.)(ENDFRM}
```

CALLED BY: CFAST, REPORTG, SOLVE

```
DEFAULT)0)LISTDEF))MESS
        *
        /)MESSNR>CSPACE))(PRINTS}
        *
        /)READMN (See Utility Subtree Section A.8)
        *
        /)MESS
        *
        /)READASTU
        *
        /)SSTRNGP
        *
        /)LENGTH
        *
        .)(ENDFRM}
```

CALLED BY: WHICHONE

```
LISTOP)0)CLEARS))(ERASE}
       *
       /)MESS
       *
       .)LISTDEF))MESS
```

CALLED BY: WHICHONE

```
PLOT2)0)WINDOW>CSPACE (See Device Subtree Section A.10)
      *
      /)CHRMOV>CSPACE))(LOCATE}
      *
      .)MESSNR>CSPACE))(PRINTS}
```

CALLED BY: WR_FLOW

```
PLOTITT)0)WINDOW>CSPACE (See Device Subtree Section A.10)
        *
        /)CHRMOV>CSPACE))(LOCATE}
        *
        .)MESSNR>CSPACE))(PRINTS}
```

CALLED BY: DISFIR, DO_CVENT

```
PRNTLIST)0)MESS
         *
         .)CLEARS))(ERASE}
```

CALLED BY: PRNTVALS, SAVE, WHICHONE

```
PRNTVALS)0)PRNTLIST (See subtree)
         *
         /)MESS
         *
         /)MESSNR>CSPACE))(PRINTS}
         *
         /)READIT (See Utility Subtree Section A.8)
         *
         .)CLEARS))(ERASE}
```

CALLED BY: WHICHONE

```
PLOTITG))GRAFIT2)0)AXSCAL)0)AUTOSC
                  *         *
                  *         /)SETAX)0)MESS
                  *         *       *
                  *         *       .)SETAXIN (See subtree)
                  *         *
                  *         .)(ENDFRM}
                  *
                  /)GETLEG)0)PUTLEG
                  *         *
                  *         /)LENGTH
                  *         *
                  *         /)MESS
                  *         *
                  *         /)MESSNR>CSPACE))(PRINTS}
                  *         *
                  *         /)(WDLEN}
                  *         *
                  *         .)SETAXIN (See subtree)
                  *
                  /)(MOUSE_OFF}
                  *
                  /)(DEVICE}
                  *
                  /)MESS
                  *
                  /)(NEWFRM}
                  *
                  /)(GRDOFF}
                  *
                  /)(LINTYP}
                  *
                  /)(LINWID}
                  *
                  /)(COLOR}
                  *
                  /)(GRISET}
                  *
                  /)(ERASE}
                  *
                  /)(GRAFIT}
                  *
                  /)(CHRSIZ}
                  *
                  /)(PLOTLN}
                  *
                  /)(WDDRAW}
                  *
                  /)(FRAME}
                  *
                  /)READASTU
                  *
                  .)(ENDFRM}
```

CALLED BY: WHICHONE

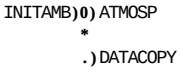
```
SETAXIN)0)LENGTH
        *
        /)MESSNR>CSPACE))(PRINTS}
        *
        .)READMN (See Utility Subtree Section A.8)
```

CALLED BY: GETLEG, SETAX

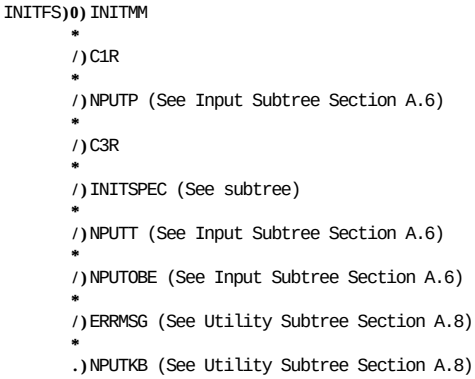
```
WHICHONE)0)(ENDFRM}
         *
         /)MESS
         *
         /)MESSNR>CSPACE))(PRINTS}
         *
         /)READASTU
         *
         /)SSTRNGP
         *
         /)LISTOP (See subtree)
         *
         /)LOWERC
         *
         /)READIT (See Utility Subtree section A.8)
         *
         /)PRNTLIST (See subtree)
         *
         /)READPLOT)0)READASTU
         *          *
         *          /)READCV1 (See Utility Subtree Section A.8)
         *          *
         *          .)MESS
         *
         /)LUNITS))MESS
         *
         /)PLOTITG (See subtree)
         *
         /)(HDCOPY}
         *
         /)READCV1 (See Utility Subtree Section A.8)
         *
         /)IOWAIT>CSPACE))(WAIT}
         *
         /)DEFAULT (See subtree)
         *
         /)SAVE (See Output Subtree Section A.7)
         *
         .)PRNTVALS (See subtree)
```

CALLED BY: CPLOT

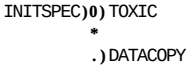
A.5 Initialization Subtrees



CALLED BY: ADD_COMP, DISAMB, NPUTP



CALLED BY: CEDIT



CALLED BY: CFAST, INITFS, REPORTG

A.6 Input Subtrees

```
DORAPID)0)MESSNR>CSPACE))(PRINTS}
        *
        /)READASTU
        *
        /)SSTRNGP
        *
        /)READIT (See Utility Subtree Section A.8)
        *
        .)MESS
```

CALLED BY: CPLOT

```
DOTENAB)0)MESSNR>CSPACE))(PRINTS}
        *
        /)READASTU
        *
        /)SSTRNGP
        *
        /)RDPRSN)0)MESSNR>CSPACE))(PRINTS}
        *         *
        *         /)READASTU
        *         *
        *         /)READCV1 (See Utility Subtree Section A.8)
        *         *
        *         /)CKDUP))BUBBLE
        *         *
        *         .)MESS
        *
        .)MESS
```

CALLED BY: CPLOT

```
GETOBJ)0)OBJDFLT
       *
       /)SETRDAS>READAS
       *
       /)READAS
       *
       /)READIN (See Utility Subtree Section A.8)
       *
       /)GTOBST)0)READAS
       *         *
       *         .)READIN (See Utility Subtree Section A.8)
       *
       .)READFL>READIN))SSTRNG
```

CALLED BY: DIS_OBJ, NPUTOB, OBTYPE

```
NPUTG)0)READCV (See Utility Subtree Section A.8)                                    *
      /)LOADUP))READCV (See Utility Subtree Section A.8)
      *
      /)GASLOAD))READCV (See Utility Subtree Section A.8)
      *
      .)READAS
```

CALLED BY: NPUTP(CFAST, REPORTG)

```
NPUTG)0)READCV (See Utility Subtree Section A.8)
      *
      .)READAS
```

CALLED BY: NPUTP(CEDIT)

```
NPUTOB)0)OPENOBJ (See subtree)
       *
       /)OBJFND
       *
       /)OBTYPE))GETOBJ (See subtree)
       *
       .)GETOBJ (See subtree)
```

CALLED BY: CFAST

```
NPUTOBE)0)OPENOBJ (See subtree)
        *
        /)OBJFND
        *
        /)OBTYPE))GETOBJ (See subtree)
        *
        .)OBJSHFL
```

CALLED BY: CEDIT, NEW_OBJ

```
NPUTP)0)NPUTQ)0)READBF>READIN0)READAS
      *       *              *
      *       *              .)SSTRNG
      *       *
      *       /)READIN (See Utility Subtree Section A.8)
      *       *
      *       /)READRS>READIN
      *       *
      *       .)READFL>READIN))SSTRNG
      *
      /)OFFSET
      *
      /)INITAMB (See Initialization Subtree Section A.5)
      *
      /)HVINIT))HVMAP
      *
      .)NPUTG (See subtree by main module CEDIT or CFAST, REPORTG)
```

CALLED BY: CFAST, INITFS, REPORTG

```
NPUTT)0)READAS
      *
      /)SSTRNG
      *
      .)READIN (See Utility Subtree Section A.8)
```

CALLED BY: CFAST, INITFS, NEWTHRM

```
OPENOBJ)0)READFL>READIN))SSTRNG
        *
        /)READIN (See Utility Subtree Section A.8)
        *
        .)OBTYPE))GETOBJ (See subtree)
```

CALLED BY: NPUTOB, NPUTOBE

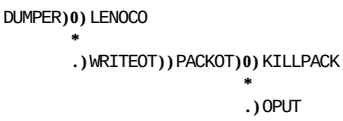
```
READHI)0)OPENFILE)0)LENOCO
       *          *
       *          /)DREADIN))UNPACK
       *          *
       *          .)F_VERS>F_INFO
       *
       /)DREADIN))UNPACK
       *
       /)F_TIME>F_INFO))FINFOT16>F_INFO16
       *
       /)DATACOPY
       *
       .)MESS
```

CALLED BY: CPLOT, REPORT, REPORTG

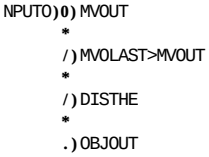
```
RESTRT)0)LENOCO
       *
       /)DREADIN))UNPACK
       *
       .)DATACOPY
```

CALLED BY: CFAST

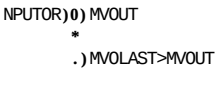
A.7 Output Subtrees



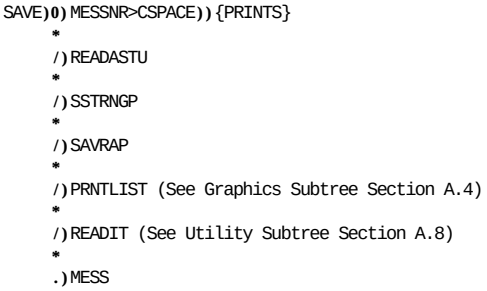
CALLED BY: CFAST, SOLVE



CALLED BY: CFAST



CALLED BY: REPORT



CALLED BY: WHICHONE

```
WRITEOF)0)WRT_GEO
        *
        /)F_FIT
        *
        /)F_FAN
        *
        /)WRT_THP
        *
        /)ALLZRO
        *
        .)GTRAILER
```

CALLED BY: QUITFI, RUNFILE

A.8 Utility Subtrees

```
CHECKI)0)CONVRT))DATYPE
       *
       .)ERRMSG (See subtree)
```

CALLED BY: ADD_CONN, ADD_DUCT, ADD_FAN, ADD_OBJ, ADD_VENT, ADD_VVNT, DISFAN, DISFIR, DISGEN, DISOB1

```
CHECKR)0)CONVRT))DATYPE
       *
       .)ERRMSG (See subtree)
```

CALLED BY: ADD_COMP, ADD_CONN, ADD_DUCT, ADD_FAN, ADD_OBJ, ADD_VENT, ADD_VVNT, DISAMB, DISDCT, DISFAN, DISFIR, DISGEO, DISOB1, DISVT1, DISVT2, DISVT3, DO_CVENT, MODTIM

```
ERRMSG)0)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       .)MESSNS (See subtree)
```

CALLED BY: ADD_COMP, ADD_CONN, ADD_DUCT, ADD_FAN, ADD_OBJ, ADD_VENT, ADD_VVNT, CEDIT, CHECKA, CHECKI, CHECKR, DEL_COMP, DEL_CONN, DEL_DUCT, DEL_FAN, DEL_OBJ, DEL_VENT, DEL_VVNT, DISDCT, DISFAN, DISFIR, DISGEO, DISOB1, DISOB2, DISTH1, DISTH2, DISVT1, DISVT2, DISVT3, MODTIM, NEW_OBJ, NEWFILE, NEWTHRM, NWHCNT, NWVCNT, OUTFIL, QUITFI, RUNFILE, WR_FLOW

```
FFILE)0)FILEFRST>CSPACE)0){FINDFRST}
      *                  *
      *                  .)SSTRNG
      *
      /)SSTRNG
      *
      /)FILENEXT>CSPACE)0){FINDNEXT}
      *                  *
      *                  .)SSTRNG
      *
      /)SWINDOW))(CAPTSCN}
      *
      /)MESSNS (See subtree)
      *
      /)WINDOW>CSPACE (See Device Subtree Section A.10)
      *
      /)FILESORT
      *
      /)CHRMOV>CSPACE))(LOCATE}
      *
      /)MESSNR>CSPACE))(PRINTS}
      *
      /)BARCODE>CSPACE (See Device Subtree Section A.10)
      *
      /)NPUTKB (See subtree)
      *
      /)RWINDOW>SWINDOW))(DISPSCN}
      *
      /)SCROLLU>CSPACE (See Device Subtree Section A.10)
      *
      .)SCROLLD>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: NEWFILE

```
GOPAGE)0)SWINDOW))(CAPTSCN}
       *
       /)MESSNS (See subtree)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)NPUTKB (See subtree)
       *
       /)RWINDOW>SWINDOW))(DISPSCN}
       *
       .)HELP (See subtree)
```

CALLED BY: DISAMB, DISCAL, DISFIR, DISGEN, DISGEO, DISOB1, DISPRM, DISTH1, DISVT1, DISVT2, DISVT3

```
HEADING)0)TOPSCR))WINDOW>CSPACE (See Device Subtree Section)
       *
       .)MESSNS (See subtree)
```

CALLED BY: DISAMB, DISCAL, DISDCT, DISFAN, DISFIR, DISGEN, DISGEO, DISOB1, DISOB2, DISPRM, DISTH1, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, MODTIM, NEWFILE, OUTFIL, PRMCLR, SUNITS

```
HELP)0)HLPTXT (See subtree)
     *
     /)NPUTKB (See subtree)
     *
     /)RWINDOW>SWINDOW))(DISPSCN}
     *
     /)HLPXTND)0)SWINDOW))(CAPTSCN}
     *         *
     *         /)WINDOW>CSPACE (See Device Subtree Section A.10)
     *         *
     *         /)CHRMOV>CSPACE))(LOCATE}
     *         *
     *         /)SETCLR
     *         *
     *         /)MESSNR>CSPACE))(PRINTS}
     *         *
     *         /)TOUPPER
     *         *
     *         /)TOLOWER>TOUPPER
     *         *
     *         .)HLPTXT (See subtree)
     *
     .)SSTRNG
```

CALLED BY: DISAMB, DISCAL, DISDCT, DISFAN, DISFIR, DISGEN, DISGEO, DISOB1, DISOB2, DISPRM, DISTH1, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, GOPAGE, MODTIM, NEW_OBJ, NEWFILE, NEWTHRM, OUTFIL, RUNFILE, SUNITS

```
HLPINS)0)OPENHELP
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)CHRMOV>CSPACE))(LOCATE}
       *
       .)MESSNR>CSPACE))(PRINTS}
```

CALLED BY: ADD_COMP, ADD_CONN, ADD_DUCT, ADD_FAN, ADD_OBJ, ADD_VENT, ADD_VVNT, DEL_COMP, DEL_CONN, DEL_DUCT, DEL_FAN, DEL_OBJ, DEL_VENT, DEL_VVNT, DISPRM, NEW_OBJ, OUTFIL, PRMCLR, QUITFI, SUNITS

```
HLPTXT)0)OPENHELP
       *
       /)SWINDOW))(CAPTSCN}
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)CHRMOV>CSPACE))(LOCATE}
       *
       .)MESSNR>CSPACE))(PRINTS}
```

CALLED BY: DISCAL, HELP, HLPXTND, TELLKEYS

```
LSTVAL)0)MESSNS (See subtree)
       *
       /)OUTI))MESSNS (See subtree)
       *
       .)OUTR))MESSNS (See subtree)
```

CALLED BY: DISGEO, DISOB1, DISTH1, DISVT1, DISVT2, DISVT3

```
MESSNS)0)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)CHRMOV>CSPACE))<LOCATE>
       *
       .)MESSNR>CSPACE))<PRINTS>
```

CALLED BY: ADD_CONN, ADD_OBJ, ADD_VVNT, CEDIT, DIS_THRM, DIS_OBJ, DISAMB, DISCAL, DISDCT, DISFAN, DISFIR, DISGEN, DISOB1, DISOB2, DISPRM, DISTH1, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, ERRMSG, FFILE, GOPAGE, HEADING, LSTFIL, LSTVAL, MODTIM, NEW_OBJ, NEWFILE, NPUTKB, OUTFIL, OUTI, OUTR, PRMCLR, RANGE, SUNITS, WR_COEF, WR_CONN, WR_DUCT, WR_FAN, WR_FLOW, WR_OBJ, WR_VENT, WR_VVNT

```
NEWFILE)0)HEADING (See subtree)
        *
        /)MESSNS (See subtree)
        *
        /)NPUTKB (See subtree)
        *
        /)SSTRNG
        *
        /)ERRMSG (See subtree)
        *
        /)HELP (See subtree)
        *
        /)MESSNR>CSPACE))<PRINTS>
        *
        /)FFILE (See subtree)
        *
        /)TELLKEYS (See subtree)
        *
        /)RMOUSE>CSPACE (See Device Subtree Section A.10)
        *
        .)CSPACE (See Device Subtree Section A.10)
```

CALLED BY: CEDIT

```
NPUTKB)0)BOTSCR>TOPSCR))WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)CHRMOV>CSPACE))(LOCATE)
       *
       /)MESSNR>CSPACE))(PRINTS)
       *
       /)SETCLR
       *
       /)RSMS))CMOUSE>CSPACE))(INTRUP)
       *
       /)MESSNS (See subtree)
       *
       /)CURSON>CSPACE))(CURON)
       *
       /)CURSOF>CSPACE))(CUROFF)
       *
       /)READKB (See subtree)
       *
       .)ERSMSG>ERRMSG))WINDOW>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: ADD_COMP, ADD_CONN, ADD_DUCT, ADD_FAN, ADD_OBJ, ADD_VENT, ADD_VVNT, DEL_COMP, DEL_CONN, DEL_DUCT, DEL_FAN, DEL_OBJ, DEL_VENT, DEL_VVNT, DISAMB, DISCAL, DISDCT, DISFAN, DISFIR, DISGEN, DISGEO, DISOB1, DISOB2, DISPRM, DISTH1, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, FFILE, GOPAGE, HELP, INITFS, MODTIM, NEW_OBJ, NEWFILE, NEWTHRM, OUTFIL, PRMCLR, QUITFI, RUNFILE, SUNITS, TELLKEYS

```
OPENSHEL)0)READCF
         *
         .)SSTRNG
```

CALLED BY: CEDIT, CFAST, CPLOT, REPORT, REPORTG

```
QUITFI)0)ERRMSG (See subtree)
       *
       /)WRITEOF (See Output Subtree Section A.7)
       *
       /)CSPACE (See Device Subtree Section A.10)
       *
       /)HLPINS (See subtree)
       *
       /)NPUTKB (See subtree)
       *
       /)SSTRNG
       *
       /)TOUPPER
       *
       /)MAKECF
       *
       .)RMOUSE>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: DISAMB, DISCAL, DISFIR, DISGEN, DISGEO, DISOB1, DISPRM, DISTH1, DISVT1, DISVT2, DISVT3

```
RANGE)0)WINDOW>CSPACE (See Device Subtree Section A.10)
      *
      /)CHRMOV>CSPACE))(LOCATE}
      *
      /)MESSNR>CSPACE))(PRINTS}
      *
      /)SETCLR
      *
      /)OUTI))MESSNS (See subtree)
      *
      /)MESSNS (See subtree)
      *
      /)OUTR))MESSNS (See subtree)
      *
      .)CUNITS
```

CALLED BY: ADD_COMP, ADD_CONN, ADD_DUCT, ADD_FAN, ADD_OBJ, ADD_VENT, ADD_VVNT, DISAMB, DISDCT, DISFAN, DISFIR, DISGEN, DISGEO, DISOB1, DISTH1, DISVT1, DISVT2, DISVT3, MODTIM

```
READCV)0)SSTRNG
       *
       .)CONVRT))DATYPE
```

CALLED BY: GASLOAD, LOADUP, NPUTG, NPUTGE

```
READCV1)0)SSTRNGP
        *
        .)CONVRT))DATYPE
```

CALLED BY: RDPRSN, READIT, READMN, READPLOT, WHICHONE

```
READIN0)SSTRNG
      *
      .)CONVRT))DATYPE
```

CALLED BY: GETOBJ, GTOBST, NPUTQ, NPUTT, OPENOBJ

```
READIT)0)READASTU
       *
       .)READCV1 (See subtree)
```

CALLED BY: DORAPID, PRNTVALS, SAVE, WHICHONE

```
READKB)0)GRABKY>CSPACE (See Device Subtree Section A.10)
       *
       /)CHRMOV>CSPACE))(LOCATE}
       *
       /)MESSNR>CSPACE))(PRINTS}
       *
       .)GRABMS)0)CMOUSE>CSPACE))(INTRUP}
                 *
                 /)CKBUTS))CMOUSE>CSPACE))(INTRUP}
                 *
                 .)WAIT100>CSPACE
```

CALLED BY: NPUTKB

```
READMN)0)READASTU
       *
       .)READCV1 (See subtree)
```

CALLED BY: DEFAULT, SETAXIN

```
READOP)0)CMDLINE)0)(GETCL}
       *          *
       *          .)BSTRNG
       *
       *
       *
       /)(DATE}
       *
       /)TOUPPER
       *
       .)SSTRNG
```

```
READOP)0)CMDLINE))GETCL)0)(IARGC}
       *                  *
       *                  /)(GETARG}
       *                  *
       *                  .)SSTRNG
       *
       /)(IDATE}
       *
       /)TOUPPER
       *
       .)SSTRNG
```

CALLED BY: CEDIT, CFAST, CPLOT, REPORT, REPORTG

```
SUNITS)0)HEADING (See subtree)
       *
       /)MESSNS (See subtree)
       *
       /)HLPINS (See subtree)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)NPUTKB (See subtree)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)HELP (See subtree)
       *
       .)SET_UNT>SETUNT
```

CALLED BY: DISAMB, DISDCT, DISFAN, DISFIR, DISGEN, DISGEO, DISOB1, DISOB2, DISPRM, DISTH2, DISVT1, DISVT2, DISVT3, MODTIM

```
TELLKEYS)0)HLPTXT (See subtree)
         *
         /)NPUTKB (See subtree)
         *
         .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISAMB, DISDCT, DISFAN, DISFIR, DISGEN, DISGEO, DISOB1, DISOB2, DISTH1, DISTH2, DISVT1, DISVT2, DISVT3, NEWFILE

A.9 Display Subtrees

```
ADD_COMP)0)SWINDOW))(CAPTSCN}
         *
         /)HLPINS (See Utility Subtree Section A.8)
         *
         /)SETCLR
         *
         /)OUTI))MESSNS (See Utility Subtree Section A.8)
         *
         /)SET_PAGE>POSIT
         *
         /)POSIT
         *
         /)POSPAGE>POSIT
         *
         /)RANGE (See Utility Subtree Section A.8)
         *
         /)CUNITS
         *
         /)BARCODE>CSPACE (See Device Subtree Section A.10)
         *
         /)NPUTKB (See Utility Subtree Section A.8)
         *
         /)SSTRNG
         *
         /)CHECKR (See Utility Subtree Section A.8)
         *
         /)OUTR))MESSNS (See Utility Subtree Section A.8)
         *
         /)ERRMSG (See Utility Subtree Section A.8)
         *
         /)INITAMB (See Initialization Subtree Section A.5)
         *
         /)RWINDOW>SWINDOW))(DISPSCN}
         *
         .)CALC_OFF>DISCAL
```

CALLED BY: DISGEO

```
ADD_CONN)0)SWINDOW))(CAPTSCN}
          *
          /)HLPINS (See Utility Subtree Section A.8)
          *
          /)SETCLR
          *
          /)SET_PAGE>POSIT
          *
          /)POSIT
          *
          /)POSPAGE>POSIT
          *
          /)CUNITS
          *
          /)RANGE (See Utility Subtree Section A.8)
          *
          /)BARCODE>CSPACE (See Device Subtree Section A.10)
          *
          /)NPUTKB (See Utility Subtree Section A.8)
          *
          /)SSTRNG
          *
          /)CHECKR (See Utility Subtree Section A.8)
          *
          /)OUTR))MESSNS (See Utility Subtree Section A.8)
          *
          /)ERRMSG (See Utility Subtree Section A.8)
          *
          /)CHECKI (See Utility Subtree Section A.8)
          *
          /)OUTI))MESSNS (See Utility Subtree Section A.8)
          *
          /)CHECKA))ERRMSG (See Utility Subtree Section A.8)
          *
          /)MESSNS (See Utility Subtree Section A.8)
          *
          /)TOUPPER
          *
          .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISVT2

```
ADD_DUCT)0)SWINDOW))(CAPTSCN}
          *
          /)HLPINS (See Utility Subtree Section A.8)
          *
          /)SETCLR
          *
          /)OUTR))MESSNS (See Utility Subtree Section A.8)
          *
          /)SET_PAGE>POSIT
          *
          /)POSIT
          *
          /)POSPAGE>POSIT
          *
          /)CUNITS
          *
          /)RANGE (See Utility Subtree Section A.8)
          *
          /)ERRMSG (See Utility Subtree Section A.8)
          *
          /)BARCODE>CSPACE (See Device Subtree Section A.10)
          *
          /)NPUTKB (See Utility Subtree Section A.8)
          *
          /)SSTRNG
          *
          /)CHECKR (See Utility Subtree Section A.8)
          *
          /)CHECKI (See Utility Subtree Section A.8)
          *
          /)OUTI))MESSNS (See Utility Subtree Section A.8)
          *
          .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISDCT

```
ADD_FAN)0)SWINDOW))[CAPTSCN]
        *
        /)HLPINS (See Utility Subtree Section A.8)
        *
        /)SETCLR
        *
        /)SET_PAGE>POSIT
        *
        /)POSIT
        *
        /)POSPAGE>POSIT
        *
        /)RANGE (See Utility Subtree Section A.8)
        *
        /)CUNITS
        *
        /)ERRMSG (See Utility Subtree Section A.8)
        *
        /)BARCODE>CSPACE (See Device Subtree Section A.10)
        *
        /)NPUTKB (See Utility Subtree Section A.8)
        *
        /)SSTRNG
        *
        /)CHECKR (See Utility Subtree Section A.8)
        *
        /)OUTR))MESSNS (See Utility Subtree Section A.8)
        *
        /)CHECKI (See Utility Subtree Section A.8)
        *
        /)OUTI))MESSNS (See Utility Subtree Section A.8)
        *
        .)RWINDOW>SWINDOW))[DISPSCN]
```

CALLED BY: DISFAN

```
ADD_OBJ))0))SWINDOW)))){CAPTSCN}
        *
        /)HLPINS (See Utility Subtree Section A.8)
        *
        /)SETCLR
        *
        /)SET_PAGE>POSIT
        *
        /)POSIT
        *
        /)POSPAGE>POSIT
        *
        /)MESSNS (See Utility Subtree Section A.8)
        *
        /)RANGE (See Utility Subtree Section A.8)
        *
        /)CUNITS
        *
        /)BARCODE>CSPACE (See Device Subtree Section A.10)
        *
        /)NPUTKB (See Utility Subtree Section A.8)
        *
        /)SSTRNG
        *
        /)CHECKR (See Utility Subtree Section A.8)
        *
        /)OUTR))MESSNS (See Utility Subtree Section A.8)
        *
        /)ERRMSG (See Utility Subtree Section A.8)
        *
        /)CHECKI (See Utility Subtree Section A.8)
        *
        /)OUTI))MESSNS (See Utility Subtree Section A.8)
        *
        /)CHECKA))ERRMSG (See Utility Subtree Section A.8)
        *
        /)OBJFND
        *
        /)OBTYPE))GETOBJ (See Input Subtree Section A.6)
        *
        /)RWINDOW>SWINDOW)))){DISPSCN}
        *
        .)DISOB2 (See subtree)
```

CALLED BY: DISOB1

```
ADD_VENT)0)SWINDOW))(CAPTSCN}
          *
          /)HLPINS (See Utility Subtree Section A.8)
          *
          /)SETCLR
          *
          /)SET_PAGE>POSIT
          *
          /)POSIT
          *
          /)POSPAGE>POSIT
          *
          /)RANGE (See Utility Subtree Section A.8)
          *
          /)CUNITS
          *
          /)BARCODE>CSPACE (See Device Subtree Section A.10)
          *
          /)NPUTKB (See Utility Subtree Section A.8)
          *
          /)SSTRNG
          *
          /)CHECKR (See Utility Subtree Section A.8)
          *
          /)OUTR))MESSNS (See Utility Subtree Section A.8)
          *
          /)ERRMSG (See Utility Subtree Section A.8)
          *
          /)CHECKI (See Utility Subtree Section A.8)
          *
          /)OUTI))MESSNS (See Utility Subtree Section A.8)
          *
          /)UP_VENT
          *
          .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISVT1

```
ADD_VVNT)0)SWINDOW))(CAPTSCN}
          *
          /)HLPINS (See Utility Subtree Section A.8)
          *
          /)SETCLR
          *
          /)SET_PAGE>POSIT
          *
          /)POSIT
          *
          /)POSPAGE>POSIT
          *
          /)WINDOW>CSPACE (See Device Subtree Section A.10)
          *
          /)MESSNS (See Utility Subtree Section A.8)
          *
          /)CUNITS
          *
          /)RANGE (See Utility Subtree Section A.8)
          *
          /)BARCODE>CSPACE (See Device Subtree Section A.10)
          *
          /)NPUTKB (See Utility Subtree Section A.8)
          *
          /)SSTRNG
          *
          /)CHECKR (See Utility Subtree Section A.8)
          *
          /)OUTR))MESSNS (See Utility Subtree Section A.8)
          *
          /)ERRMSG (See Utility Subtree Section A.8)
          *
          /)CHECKI (See Utility Subtree Section A.8)
          *
          /)OUTI))MESSNS (See Utility Subtree Section A.8)
          *
          /)CHECKA))ERRMSG (See Utility Subtree Section A.8)
          *
          .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISVT3

```
DEL_COMP)0)SWINDOW))(CAPTSCN}
          *
          /)HLPINS (See Utility Subtree Section A.8)
          *
          /)NPUTKB (See Utility Subtree Section A.8)
          *
          /)SSTRNG
          *
          /)TOUPPER
          *
          /)ERRMSG (See Utility Subtree Section A.8)
          *
          /)RWINDOW>SWINDOW))(DISPSCN}
          *
          .)CALC_OFF>DISCAL
```

CALLED BY: DISGEO

```
DEL_CONN)0)SWINDOW))(CAPTSCN}
         *
         /)HLPINS (See Utility Subtree Section A.8)
         *
         /)NPUTKB (See Utility Subtree Section A.8)
         *
         /)SSTRNG
         *
         /)TOUPPER
         *
         /)ERRMSG (See Utility Subtree Section A.8)
         *
         .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISVT2

```
DEL_DUCT)0)SWINDOW))(CAPTSCN}
         *
         /)HLPINS (See Utility Subtree Section A.8)
         *
         /)NPUTKB (See Utility Subtree Section A.8)
         *
         /)SSTRNG
         *
         /)TOUPPER
         *
         /)ERRMSG (See Utility Subtree Section A.8)
         *
         /)RWINDOW>SWINDOW))(DISPSCN}
         *
         .)F_FIT
```

CALLED BY: DISDCT

```
DEL_FAN)0)SWINDOW))(CAPTSCN}
        *
        /)HLPINS (See Utility Subtree Section A.8)
        *
        /)NPUTKB (See Utility Subtree Section A.8)
        *
        /)SSTRNG
        *
        /)TOUPPER
        *
        /)ERRMSG (See Utility Subtree Section A.8)
        *
        /)RWINDOW>SWINDOW))(DISPSCN}
        *
        .)F_FAN
```

CALLED BY: DISFAN

```
DEL_OBJ)0)SWINDOW))(CAPTSCN}
        *
        /)HLPINS (See Utility Subtree Section A.8)
        *
        /)NPUTKB (See Utility Subtree Section A.8)
        *
        /)SSTRNG
        *
        /)ERRMSG (See Utility Subtree Section A.8)
        *
        /)RWINDOW>SWINDOW))(DISPSCN}
        *
        /)OBTYPE))GETOBJ (See Input Subtree Section A.6)
        *
        .)OBJSHFL
```

CALLED BY: DISOB1

```
DEL_VENT)0)SWINDOW))(CAPTSCN}
         *
         /)HLPINS (See Utility Subtree Section A.8)
         *
         /)NPUTKB (See Utility Subtree Section A.8)
         *
         /)SSTRNG
         *
         /)TOUPPER
         *
         /)ERRMSG (See Utility Subtree Section A.8)
         *
         .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISVT1

```
DEL_VVNT)0)SWINDOW))(CAPTSCN}
         *
         /)HLPINS (See Utility Subtree Section A.8)
         *
         /)NPUTKB (See Utility Subtree Section A.8)
         *
         /)SSTRNG
         *
         /)ERRMSG (See Utility Subtree Section A.8)
         *
         .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISVT3

```
DISAMB)0)HEADING (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)OUTI))MESSNS (See Utility Subtree Section A.8)
       *
       /)INITAMB (See Initialization Subtree Section A.5)
       *
       /)POSIT
       *
       /)POSPAGE>POSIT
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       .)QUITFI (See Utility Subtree Section A.8)
```

CALLED BY: CEDIT

```
DISCAL)0)HEADING (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)HLPTXT (See Utility Subtree Section A.8)
       *
       /)RWINDOW>SWINDOW))){DISPSCN}
       *
       /)OUTFIL)0)HEADING (See Utility Subtree Section A.8)
       *        *
       *        /)MESSNS (See Utility Subtree Section A.8)
       *        *
       *        /)HLPINS (See Utility Subtree Section A.8)
       *        *
       *        /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *        *
       *        /)NPUTKB (See Utility Subtree Section A.8)
       *        *
       *        /)ERRMSG (See Utility Subtree Section A.8)
       *        *
       *        /)RUNFILE)0)ERRMSG (See Utility Subtree Section A.8)
       *        *         *
       *        *         /)NPUTKB (See Utility Subtree Section A.8)
       *        *         *
       *        *         /)SSTRNG
       *        *         *
       *        *         /)HELP (See Utility Subtree Section A.8)
       *        *         *
       *        *         .)WRITEOF (See Output Subtree Section A.7)
       *        *
       *        /)SSTRNG
       *        *
       *        .)HELP (See Utility Subtree Section A.8)
       *
       /){TIME}
       *
       .)QUITFI (See Utility Subtree Section A.8)
```

CALLED BY: CEDIT

```
DISDCT)0)POSIT
       *
       /)HEADING (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)WR_DUCT)0)CUNITS
       *         *
       *         /)F_FIT
       *         *
       *         .)MESSNS (See Utility Subtree Section A.8)
       *
       /)POSPAGE>POSIT
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)F_FIT
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)ADD_DUCT (See subtree)
       *
       /)DEL_DUCT (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: DISVT2

```
DISFAN)0)HEADING (See Utility Subtree Section A.8)
       *
       /)POSIT
       *
       /)SETCLR
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)CALC_FAN
       *
       /)WR_FAN)0)MESSNS (See Utility Subtree Section A.8)
       *        *
       *        /)F_FAN
       *        *
       *        /)CUNITS
       *        *
       *        .)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)WR_FLOW)0)CUNITS
       *         *
       *         /)OUTR6>OUTR
       *         *
       *         /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *         *
       *         /)MESSNS (See Utility Subtree Section A.8)
       *         *
       *         /)PLOT2 (See Graphics Subtree Section A.4)
       *         *
       *         .)ERRMSG (See Utility Subtree Section A.8)
       *
       /)WR_COEF)0)CF_UNT
       *         *
       *         .)MESSNS (See Utility Subtree Section A.8)
       *
       /)POSPAGE>POSIT
       *
       /)CUNITS
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)REGRESS
       *
       /)CHECKI (See Utility Subtree Section A.8)
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)OUTR6>OUTR
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)CF_UNT
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)ADD_FAN (See subtree)
       *
       /)DEL_FAN (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: DISVT2

```
DISFIR)0)HEADING (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)OUTI))MESSNS (See Utility Subtree Section A.8)
       *
       /)POSIT
       *
       /)POSPAGE>POSIT
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)PLOTITT (See Graphics Subtree Section A.4)
       *
       /)CHRMOV>CSPACE))(LOCATE}
       *
       /)MESSNR>CSPACE))(PRINTS}
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)CHECKI (See Utility Subtree Section A.8)
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)UP_FIRE
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)MODTIM (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)QUITFI (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: CEDIT

```
DISGEN)0)HEADING (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)OUTI))MESSNS (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)POSIT
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)POSPAGE>POSIT
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)TURNKBON>NPUTKB
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)SSTRNG
       *
       /)CHECKI (See Utility Subtree Section A.8)
       *
       /)CHECKA))ERRMSG (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       .)QUITFI (See Utility Subtree Section A.8)
```

CALLED BY: CEDIT

```
DISGEO)0)HEADING (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)LSTVAL (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)POSIT
       *
       /)POSPAGE>POSIT
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)ADD_COMP (See subtree)
       *
       /)DEL_COMP (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       .)QUITFI (See Utility Subtree Section A.8)
```

CALLED BY: CEDIT

```
DISOB1)0)POSIT
       *
       /)NWOCNT
       *
       /)HEADING (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)LSTVAL (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)WR_OBJ)0)CUNITS
       *        *
       *        .)MESSNS (See Utility Subtree Section A.8)
       *
       /)POSPAGE>POSIT
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)SSTRNG
       *
       /)CHECKA))ERRMSG (See Utility Subtree Section A.8)
       *
       /)OBJFND
       *
       /)OBTYPE))GETOBJ (See Input Subtree Section A.6)
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)CHECKI (See Utility Subtree Section A.8)
       *
       /)OUTI))MESSNS (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)NEW_OBJ (See subtree)
       *
       /)ADD_OBJ (See subtree)
       *
       /)DEL_OBJ (See subtree)
       *
       /)DISOB2 (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)QUITFI (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: CEDIT

```
DISOB2)0)SWINDOW))(CAPTSCN}
       *
       /)HEADING (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)DIS_OBJ)0)GETOBJ (See Input Subtree Section A.6)
       *          *
       *          /)CUNITS
       *          *
       *          .)MESSNS (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)OBTYPE))GETOBJ (See Input Subtree Section A.6)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       /)SCROLLU>CSPACE (See Device Subtree Section A.10)
       *
       .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: ADD_OBJ, DISOB1

```
DISPRM)0)HEADING (See Utility Subtree Section A.8)
       *
       /)HLPINS (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)PRMCLR)0)HEADING (See Utility Subtree Section A.8)
       *         *
       *         /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *         *
       *         /)MESSNS (See Utility Subtree Section A.8)
       *         *
       *         /)HLPINS (See Utility Subtree Section A.8)
       *         *
       *         .)NPUTKB (See Utility Subtree Section A.8)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)MAKECF
       *
       .)QUITFI (See Utility Subtree Section A.8)
```

CALLED BY: CEDIT

```
DISTH1)0)HEADING (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)LSTVAL (See Utility Subtree Section A.8)
       *
       /)LSTOFF>LSTVAL
       *
       /)CUNITS
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)LSTFIL)0)MESSNS (See Utility Subtree Section A.8)
       *        *
       *        .)LSTRNG
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)SSTRNG
       *
       /)CHECKA))ERRMSG (See Utility Subtree Section A.8)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)NEWTHRM (See subtree)
       *
       /)DISTH2 (See subtree)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       .)QUITFI (See Utility Subtree Section A.8)
```

CALLED BY: CEDIT

```
DISTH2)0)SWINDOW))(CAPTSCN}
       *
       /)HEADING (See Utility Subtree Section A.8)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)DIS_THRM)0)CUNITS
       *          *
       *          /)MESSNS (See Utility Subtree Section A.8)
       *          *
       *          /)SETCLR
       *          *
       *          .)PRNVAL)0)OUTI))MESSNS (See Utility Subtree Section A.8)
       *                    *
       *                    .)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       /)SCROLLU>CSPACE (See Device Subtree Section A.10)
       *
       .)RWINDOW>SWINDOW))(DISPSCN}
```

CALLED BY: DISTH1

```
DISVT1)0)POSIT
       *
       /)NWHCNT))ERRMSG (See Utility Subtree Section A.8)
       *
       /)HEADING (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)LSTVAL (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)WR_VENT)0)CUNITS
       *         *
       *         .)MESSNS (See Utility Subtree Section A.8)
       *
       /)POSPAGE>POSIT
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)UP_VENT
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)ADD_VENT (See subtree)
       *
       /)DEL_VENT (See subtree)
       *
       /)DO_CVENT (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)QUITFI (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: CEDIT

```
DISVT2)0)POSIT
       *
       /)HEADING (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)LSTVAL (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)WR_CONN)0)CUNITS
       *         *
       *         .)MESSNS (See Utility Subtree Section A.8)
       *
       /)POSPAGE>POSIT
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)SSTRNG
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)ADD_CONN (See subtree)
       *
       /)DEL_CONN (See subtree)
       *
       /)DISDCT (See subtree)
       *
       /)DISFAN (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)QUITFI (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: CEDIT

```
DISVT3)0)POSIT
       *
       /)NWVCNT))ERRMSG (See Utility Subtree Section A.8)
       *
       /)HEADING (See Utility Subtree Section A.8)
       *
       /)SETCLR
       *
       /)LSTVAL (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)WR_VVNT)0)CUNITS
       *         *
       *         .)MESSNS (See Utility Subtree Section A.8)
       *
       /)POSPAGE>POSIT
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)SSTRNG
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)GOPAGE (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)ADD_VVNT (See subtree)
       *
       /)DEL_VVNT (See subtree)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)TELLKEYS (See Utility Subtree Section A.8)
       *
       /)QUITFI (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: CEDIT

```
DO_CVENT)0)SWINDOW))(CAPTSCN)
         *
         /)HEADING (See Utility Subtree Section A.8)
         *
         /)MESSNS (See Utility Subtree Section A.8)
         *
         /)PLOTITT (See Graphics Subtree Section A.4)
         *
         /)WINDOW>CSPACE (See Device Subtree Section A.10)
         *
         /)SETCLR
         *
         /)CHRMOV>CSPACE))(LOCATE)
         *
         /)MESSNR>CSPACE))(PRINTS)
         *
         /)OUTR))MESSNS (See Utility Subtree Section A.8)
         *
         /)BARCODE>CSPACE (See Device Subtree Section A.10)
         *
         /)NPUTKB (See Utility Subtree Section A.8)
         *
         /)CHECKR (See Utility Subtree Section A.8)
         *
         /)SCROLLD>CSPACE (See Device Subtree Section A.10)
         *
         /)HELP (See Utility Subtree Section A.8)
         *
         /)RWINDOW>SWINDOW))(DISPSCN)
         *
         .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: DISVT1

```
MODTIM)0)HEADING (See Utility Subtree Section A.8)
       *
       /)CUNITS
       *
       /)WINDOW>CSPACE (See Device Subtree Section A.10)
       *
       /)SETCLR
       *
       /)CHRMOV>CSPACE))(LOCATE)
       *
       /)MESSNR>CSPACE))(PRINTS)
       *
       /)OUTR))MESSNS (See Utility Subtree Section A.8)
       *
       /)BARCODE>CSPACE (See Device Subtree Section A.10)
       *
       /)MESSNS (See Utility Subtree Section A.8)
       *
       /)RANGE (See Utility Subtree Section A.8)
       *
       /)NPUTKB (See Utility Subtree Section A.8)
       *
       /)CHECKR (See Utility Subtree Section A.8)
       *
       /)ERRMSG (See Utility Subtree Section A.8)
       *
       /)HELP (See Utility Subtree Section A.8)
       *
       /)SUNITS (See Utility Subtree Section A.8)
       *
       /)SCROLLD>CSPACE (See Device Subtree Section A.10)
       *
       .)SCROLLU>CSPACE (See Device Subtree Section A.10)
```

CALLED BY: DISFIR

```
NEW_OBJ)0)SETCLR
        *
        /)BARCODE>CSPACE (See Device Subtree Section A.10)
        *
        /)NPUTKB (See Utility Subtree Section A.8)
        *
        /)SSTRNG
        *
        /)SWINDOW))(CAPTSCN)
        *
        /)HLPINS (See Utility Subtree Section A.8)
        *
        /)MESSNS (See Utility Subtree Section A.8)
        *
        /)ERRMSG (See Utility Subtree Section A.8)
        *
        /)RWINDOW>SWINDOW))(DISPSCN)
        *
        /)NPUTOBE (See Input Subtree Section A.6)
        *
        .)HELP (See Utility Subtree Section A.8)
```

CALLED BY: DISOB1

```
NEWTHRM)0)SETCLR
        *
        /)BARCODE>CSPACE (See Device Subtree Section A.10)
        *
        /)NPUTKB (See Utility Subtree Section A.8)
        *
        /)SSTRNG
        *
        /)NPUTT (See Input Subtree Section A.6)
        *
        /)ERRMSG (See Utility Subtree Section A.8)
        *
        .)HELP (See Utility Subtree Section A.8)
```

CALLED BY: DISTH1

A.10 Device Subtrees

```
BARCODE>CSPACE)0) {VSTOWA}
                 *
                 .) {VSHOWA}
```

CALLED BY: ADD_COMP, ADD_CONN, ADD_DUCT, ADD_FAN, ADD_OBJ, ADD_VENT, ADD_VVNT,
DISAMB, DISDCT, DISFAN, DISFIR, DISGEN, DISGEO, DISOB1, DISOB2,
DISTH1, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, FFILE, GOPAGE,
MODTIM, NEW_OBJ, NEWTHRM, OUTFIL, SUNITS

```
CSPACE)0) {VIDEOTYP}
         *
         /) {VIDADDR}
         *
         /) {SETATTR}
         *
         .) {CLS}
```

CALLED BY: CEDIT, NEWFILE, QUITFI

```
GRABKY>CSPACE)0) {KBHIT}
                *
                .) {GETCH}
```

CALLED BY: DEBUGPR, NTRACT, READKB, SLVHELP

```
IMOUSE)0) CMOUSE>CSPACE)) {INTRUP}
         *
         .) SMOUSE>CSPACE (See subtree)
```

CALLED BY: CEDIT

```
RMOUSE>CSPACE)0) {OFFSET}
                *
                /) {SEGMENT}
                *
                .) {INTRUP}
```

CALLED BY: QUITFI

```
SCROLLD>CSPACE)0) {SETATTR}
               *
               .) {SCROLL}
```

CALLED BY: DISDCT, DISFAN, DISFIR, DISOB1, DISOB2, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, FFILE, MODTIM

```
SCROLLU>CSPACE)0) {SETATTR}
               *
               .) {SCROLL}
```

CALLED BY: DISDCT, DISFAN, DISFIR, DISOB1, DISOB2, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, FFILE, MODTIM

```
SMOUSE>CSPACE)0) {OFFSET}
               *
               /) {SEGMENT}
               *
               .) {INTRUP}
```

CALLED BY: IMOUSE

```
WINDOW>CSPACE)0) {SETATTR}
               *
               .) {CLS}
```

CALLED BY: ADD_VVNT, BOTSCR, DISFIR, DISOB1, DISOB2, DISTH2, DISVT1, DISVT2, DISVT3, DO_CVENT, ERRMSG, ERSMSG, FFILE, HLPINS, HLPTXT, HLPXTND, MESSNS, MODTIM, PLOT2, PLOTITT, PRMCLR, RANGE, SUNITS, TOPSCR

Appendix B: Alphabetical Listing of CFAST Routines

NOTATION: `file1>file2` indicates an entry point `file1` within the source routine `file2`

NAME	DECLARATION	DESCRIPTION
ADD_COMP	DISGEO.SOR	Add a compartment in CEdit.
ADD_CONN	DISVT2.SOR	Add an external HVAC connection in CEdit
ADD_DUCT	DISDCT.SOR	Add a HVAC duct in CEdit.
ADD_FAN	DISFAN.SOR	Add a HVAC fan in CEdit.
ADD_OBJ	DISOB1.SOR	Add a specified object in CEdit.
ADD_VENT	DISVT1.SOR	Add a horizontal flow vent in CEdit.
ADD_VVNT	DISVT3.SOR	Add a vertical flow vent in CEdit.
AHISTS	DISPLAYC.SOR	Store the output from OUTPU1 for later output as an ASCII text line.
ALLZRO	WRITEOFE.SOR	Logical function indicating if all values of floating point array are set to zero in CEdit.
ATMOSP	ATMOSP.SOR	Floating point function to calculate the pressure, temperature and density at a specified height.
AUTOSC	AUTOSC.SOR	Automatic scaling of the CPlot axes.
AVG	NAILED.SOR	Calculate the average of two real numbers.
AXSCAL	AXSCAL.SOR	Manual scaling of the CPlot axes.
BARCODE>CSPACE	CSPACE.SOR	Toggle the color attributes for screen location to display highlight bar.
BOTSCR>TOPSCR	TOPSCR.SOR	Clear the last screen line and set foreground and background colors.
BSTRNG	BSTRNG.SOR	String search similar to SSTRNG except it allows for null arguments separated by commas.
BUBBLE	BUBBLE.SOR	Bubble sort of TENAB input in CPlot.
BXBLIT	BXBLIT.SOR	Bit obliterator - used for erasing parts of a screen in CFAST.
C1R	C1R.SOR	Generate a CEdit single compartment case.
C3R	C3R.SOR	Generate a CEdit three compartment case.

NAME	DECLARATION	DESCRIPTION
CALC_FAN	DISFAN.SOR	Calculate HEAD and FLOW arrays from specified coefficients in CEdit.
CALC_OFF>DISCAL	DISCAL.SOR	Previously used to turn calculation off in CEdit; Obsolete routine.
CEDIT	CEDIT.SOR	Input file data editor.
CEILHT	CEILHT.SOR	Calculate convective heat transfer to the uniform temperature ceiling above a fire in a parallelepiped compartment with a two-layer fire environment.
CFAST	CFAST.SOR	Main model.
CF_SET	CF_SET.SOR	Set colors, units, paths, and default database names in the HV1.CF configuration file.
CF_UNT	DISFAN.SOR	Display fan coefficients in user units or convert the user specified values to scientific units and stores in the HVBCO array.
CHECKA	CHECKA.SOR	Check validity of an entered ASCII string.
CHECKI	CHECKI.SOR	Check validity of an entered integer value.
CHECKR	CHECKR.SOR	Check validity of an entered real value.
CHEMIE	CHEMIE.SOR	Do the combustion chemistry - for plumes in both the upper and lower layers.
CHRLOC>CSPACE	CSPACE.SOR	Return current cursor position on screen.
CHRMOV>CSPACE	CSPACE.SOR	Locate cursor to specified row and column.
CHRSET>WDDRAW	WDDRAW.SOR	Change the default character set in CFAST graphics display.
CJET	CJET.SOR	Interface between RESID and CEILHT; sets up variables to pass.
CKBUTS	MOUSE.SOR	Check to see if either mouse button has been depressed since last check.
CKDUP	CKDUP.SOR	CPlot check for duplicates in TENAB person list.
CLEARS	CLEARS.SOR	Clear the screen.
CMDLINE	CMDLINE.SOR	Read and interpret the DOS command line.

NAME	DECLARATION	DESCRIPTION
CMOUSE>CSPACE	CSPACE.SOR	Return current values in mouse interrupt vector.
CNDUCT	CNDUCT.SOR	Heat conduction through objects.
CNHEAT	CNHEAT.SOR	Interface routine between RESID and CONVEC, CNDUCT for surfaces.
CONHT	NAILED.SOR	Calculate fractional effective dose due to temperature in 1 minute in REPORT.
CONT1	NAILED.SOR	Obtain data for new time step in REPORT.
CONVEC	CONVEC.SOR	Convective heat loss or gain.
CONVRT	CONVRT.SOR	Convert an ASCII string to an integer or floating point number.
COPY	NAILED.SOR	Copy one matrix to another in REPORT.
CPLOT	CPLOT.SOR	Plotting routine.
CPTIME	CPTIME.SOR	Return the total calculation (not simulation) time since the beginning.
CSPACE	CSPACE.SOR	Determine adapter, set attributes, and clear the screen in CEdit and CF_Set.
CUNITS	CUNITS.SOR	Convert from user units to scientific units or vice versa.
CURSOF>CSPACE	CSPACE.SOR	Turn the cursor off.
CURSON>CSPACE	CSPACE.SOR	Turn the cursor on.
CVHEAT	CVHEAT.SOR	Interface between RESID and CONVEC; sets up variables to pass.
D1MACH	AAUX.SOR	Used to obtain machine-dependent parameters for the local machine environment.
DATACOPY	DATACOPY.SOR	Copy the solver variables to the environment common blocks.
DATYPE	DATYPE.SOR	Determine the type of the ASCII string (integer,floating point,...).
DAXPY	BLAS.SOR	Double precision computation of Y = A*X+Y; BLAS routine.
DDAINI	DASSL.SOR	Takes a step using the backward Euler method; part of solver.

NAME	DECLARATION	DESCRIPTION
DDAJAC	DASSL.SOR	Compute the iteration matrix; part of solver.
DDANRM	DASSL.SOR	A weighted norm routine used by solver to measure the size of vectors such as the estimated error in each step.
DDASLV	DASSL.SOR	Manages the solution of the linear system arising in the Newton iteration; part of solver.
DDASSL	DASSL.SOR	Solver.
DDASTP	DASSL.SOR	Solve a system of differential algebraic equations; part of solver.
DDATRP	DASSL.SOR	Works in conjunction with routine DDASTP to approximate the solution and its derivative at a time xout; part of solver.
DDAWTS	DASSL.SOR	Set the error weight vector; part of solver.
DDOT	BLAS.SOR	Double precision inner product of double precision vectors; BLAS routine.
DEBUGPR	SOLVE.SOR	Diagnostic routine for responding to function keys in CFAST.
DEFAULT	DEFAULT.SOR	Allow the user to set CPlot default parameters for the compartment number, layer, species, vent flow destination, and character set.
DELP	DELP.SOR	Calculate the absolute hydrostatic pressures at a specified elevation in each of two adjacent compartments and the pressure difference.
DEL_COMP	DISGEO.SOR	Delete a compartment in CEdit.
DEL_CONN	DISVT2.SOR	Delete external HVAC connection in CEdit.
DEL_DUCT	DISDCT.SOR	Delete a HVAC duct in CEdit.
DEL_FAN	DISFAN.SOR	Delete a HVAC fan in CEdit.
DEL_OBJ	DISOB1.SOR	Delete a specified object in CEdit.
DEL_VENT	DISVT1.SOR	Delete a horizontal flow vent in CEdit.
DEL_VVNT	DISVT3.SOR	Delete a vertical flow vent in CEdit.
DGBFA	LINPACK.SOR	Factor a double precision band matrix by elimination; LINPACK routine.

NAME	DECLARATION	DESCRIPTION
DGBSL	LINPACK.SOR	Solve the double precision band system a*x = b or trans(a) * x = b; LINPACK routine.
DGEFA	LINPACK.SOR	Factor a double precision matrix by Gaussian elimination; LINPACK routine.
DGESL	LINPACK.SOR	Solve the double precision system a*x = b or trans(a) * x = b; LINPACK routine.
DIOB	DIOB.SOR	Display the valid objects in the current objects database; used by LIST_OBJ.
DISAMB	DISAMB.SOR	Display CEdit Ambient Conditions screen.
DISCAL	DISCAL.SOR	Display CEdit Files,... screen.
DISCLAIM	DISCLAIM.SOR	Disclaimer notice.
DISDCT	DISDCT.SOR	Display duct subpage of CEdit Fans, Ducts,... screen.
DISFAN	DISFAN.SOR	Display fan subpage of CEdit Fans, Ducts,... screen.
DISFIR	DISFIR.SOR	Display CEdit Fire Specification screen.
DISGEN	DISGEN.SOR	Display CEdit Overview screen.
DISGEO	DISGEO.SOR	Display CEdit Geometry screen.
DISOB1	DISOB1.SOR	Display CEdit Objects screen.
DISOB2	DISOB2.SOR	Display current objects database in CEdit.
DISPLAYC	DISPLAYC.SOR	Graphics output in CFAST.
DISPRM	DISPRM.SOR	Display CEdit Version and Settings screen.
DISTH1	DISTH1.SOR	Display CEdit Thermal Properties screen.
DISTH2	DISTH2.SOR	Display current thermophysical database in CEdit.
DISTHE	DISTHE.SOR	Print combined thermophysical data (DISTH1 and DISTH2) in CFAST.
DISTHENI	DISTHENI.SOR	Similar to DISTHE without initialization; used in LIST_TPP.
DISVT1	DISVT1.SOR	Display CEdit Vents(Doors,...) screen.
DISVT2	DISVT2.SOR	Display CEdit Fans, Ducts,... screen.

NAME	DECLARATION	DESCRIPTION
DISVT3	DISVT3.SOR	Display the Vents(Ceiling,...) screen in CEdit.
DIS_OBJ	DISOB2.SOR	Display one row of CEdit objects database screen.
DIS_THRM	DISTH2.SOR	Display one row of CEdit thermal properties database screen.
DJET	DJET.SOR	Physical interface routine to calculate the current rates of mass and energy flows into the layers from all door jet fires in the structure.
DJFIRE	DJFIRE.SOR	Calculate heat and combustion chemistry for a door jet fire.
DNRM2	BLAS.SOR	Euclidean length (L2 norm) of double precision vector; BLAS routine.
DOFIRE	DOFIRE.SOR	Do heat release from a fire for both main fire and objects.
DOGLEG	SNSQE.SOR	Determine the convex combination X of the Gauss-Newton and scaled gradient directions that minimizes (A*X - B) in the least squares sense; SNSQE routine.
DORAPID	DORAPID.SOR	Read channels from RAPID file in CPlot.
DOSES	NAILED.SOR	Determine hazard the person is exposed to based on layer; used in REPORT.
DOTENAB	DOTENAB.SOR	Read from TENAB history file in CPlot.
DO_CVENT	DO_CVENT.SOR	Specify CVENT parameters on fire timeline in CEdit.
DREADIN	DREADIN.SOR	Read a record (binary) of a history file. See WRITEOT.
DSCAL	BLAS.SOR	Double precision vector scale X = A*X; BLAS routine.
DUMPER	DUMPER.SOR	Write to the history file.
ENDDIS>DISPLAYC	DISPLAYC.SOR	Cleanup for CFAST graphics display.
ENORM	SNSQE.SOR	Given an N-vector X, calculate the Euclidean norm of X; SNSQE routine.
ENTRAIN	ENTRAIN.SOR	Calculate entrainment for HFLOW.

NAME	DECLARATION	DESCRIPTION
ENTRFL	ENTRFL.SOR	Low-level routine with ENTRAIN to calculate vent entrainment.
ERRMSG	ERRMSG.SOR	Write CEdit error message on error line.
ERSMSG>ERRMSG	ERRMSG.SOR	Erase CEdit message window.
FDCO	NAILED.SOR	Fractional effective dose due to CO in 1 minute in REPORT.
FDCO2	NAILED.SOR	Fractional effective dose due to CO2 in 1 minute in REPORT.
FDHCN	NAILED.SOR	Fractional effective dose due to HCN in 1 minute in REPORT.
FDJAC1	SNSQE.SOR	Compute a forward-difference approximation to the N by N Jacobian matrix associated with a specified problem of N functions in N variables; SNSQE routine.
FDO2	NAILED.SOR	Fractional effective dose due to O2 in 1 minute in REPORT.
FFILE	FFILE.SOR	Grab a list of files and select one in CEdit.
FILEFRST>CSPACE	CSPACE.SOR	Find first filename in directory matching wildcard entry on CEdit Initialization screen.
FILENEXT>CSPACE	CSPACE.SOR	Find next filename in directory; see FILEFRST.
FILESORT	FILESORT.SOR	Sort the list of available files by name and extension (simple bubble sort) in CEdit.
FIND	NAILED.SOR	Verify a particular type of keyword in REPORT; used by MATCH.
FINFOR16>F_INFO16	F_INFO.SOR	CPlot current simulation time, version 1.6.
FINFOT16>F_INFO16	F_INFO.SOR	CPlot current simulation step, version 1.6.
FIRES	FIRES.SOR	Physical interface routine to calculate the current rates of mass and energy flows into the layers from all fires in the building.
FIRPLM	FIRPLM.SOR	Calculates plume entrainment for a fire from McCaffrey's correlation.

NAME	DECLARATION	DESCRIPTION
FLOGO1	FLOGO1.SOR	Deposition of mass, enthalpy, oxygen, and other product-of-combustion flows passing between two compartments through a vertical, constant-width vent.
FLWOUT	FLWOUT.SOR	Display the flow field in CFAST.
FM1	ENTRFL.SOR FIRPLM.SOR	Used to calculate coefficients in plume entrainment (FIRPLM) routine.
FM2	ENTRFL.SOR FIRPLM.SOR	Used to calculate coefficients in plume entrainment (FIRPLM) routine.
FM3	ENTRFL.SOR FIRPLM.SOR	Used to calculate coefficients in plume entrainment (FIRPLM) routine.
FOBJFILE	FOBJFILE.SOR	Display current objects database name and open file in LIST_OBJ.
F_FAN	FIT.SOR	Given a fan number, find the corresponding branch number.
F_FIT	FIT.SOR	Given a duct number, find the corresponding fitting numbers.
F_INFO	F_INFO.SOR	Get the version, run date, etc. of a history file.
F_INFO16	F_INFO.SOR	Used by F_INFO - get CFAST 1.6 data.
F_RCOUNT>DISPLAYC	DISPLAYC.SOR	Similar to F_RCOUNT>F_INFO but in CFAST to provide compatibility.
F_RCOUNT>F_INFO	F_INFO.SOR	CPlot count of compartments (including outside).
F_TIME>F_INFO	F_INFO.SOR	CPlot current simulation step.
F_VERS>F_INFO	F_INFO.SOR	CPlot, REPORT current version number.
GASLOAD	GASLOAD.SOR	Read species keyword specifications in graphics section of input file; used by NPUTG.
GETELEV	GETELEV.SOR	Set elevation coordinates for horizontal flow vent; called by VENT.
GETLEG	GETLEG.SOR	Create and place legends on the individual graphs in CPlot.
GETN	NAILED.SOR	Get the version, run date, etc. of a history file in CPlot by calling F_INFO.

NAME	DECLARATION	DESCRIPTION
GETOBJ	GETOBJ.SOR	Get database detailed information for a specified object.
GETVAR	GETVAR.SOR	Calculate conditions for specified compartment; used by HFLOW.
GETVIEW	GETVIEW.SOR	Read a three dimensional view for display in CFAST.
GJAC	GJAC.SOR	Evaluate the Jacobian for the solver. (Not used at this time).
GOPAGE	GOPAGE.SOR	Display menu of CEdit screens to allow transfer to another screen.
GRABKY>CSPACE	CSPACE.SOR	Read character from keyboard buffer if available.
GRAFIT	GRAFIT.SOR	Initialize a graphics grid or contour diagram with a full set of labels and switch the rest of the contour graphics material to the graphics mode; modified version for CFAST of GRAFIT from DEVICE.
GRAFIT2	GRAFIT2.SOR	CPlot modified version of GRAFIT.
GRES	GRES.SOR	User-supplied subroutine to calculate the functions; SNSQE routine.
GRES2	GRES2.SOR	User-supplied subroutine to calculate the functions; SNSQE routine.
GRILAB>GRAFIT	GRAFIT.SOR	Calculate graphics grid labels.
GRILOF>GRAFIT	GRAFIT.SOR	Turn off graphics grid labels.
GRISET>GRAFIT	GRAFIT.SOR	Set axes minimum and maximum labels.
GTOBST	GTOBST.SOR	Read a line from the objects database; used with GETOBJ.
GTRAILER	GTRAILER.SOR	Write the default graphics descriptor file to the input file when selected on Files,... screen in CEdit.
HCL	HCL.SOR	Physical interface routine to do HCl deposition on wall surfaces.
HCLTRAN	HCLTRAN.SOR	Calculate the hydrogen chloride balance in the gas and on the wall surface; called by HCL.
HEADING	HEADING.SOR	Display heading for each CEdit screen.

NAME	DECLARATION	DESCRIPTION
HELP	HELP.SOR	Display help window and wait for keyword input.
HFLOW	HFLOW.SOR	Physical interface routine to calculate flow through all unforced horizontal vents.
HLPINS	HLPINS.SOR	Display CEdit help information at specified location; used for prompts.
HLPTXT	HLPTXT.SOR	Display CEdit help information centered on the screen; used for help messages.
HLPXTND	HLPXTND.SOR	Provide keyword help inside CEdit help.
HVFAN	HVFAN.SOR	Calculate mass flow rate through a fan based on a fan curve.
HVFREX	HVFREX.SOR	Update arrays and assign compartment pressures, temperatures and species to HVAC exterior nodes (from compartments)
HVFRIC	HVFRIC.SOR	Calculate duct friction factor from the Colebrook equation.
HVINIT	HVINIT.SOR	Set up the arrays needed for HVAC simulation and initialize temperatures and concentrations.
HVMAP	HVMAP.SOR	Construct mapping arrays used to map interior nodes, exterior nodes, and HVAC systems to appropriate arrays.
HVMFLO	HVMFLO.SOR	Compute the source term for mass flow in the mechanical ventilation system.
HVSFLO	HVSFLO.SOR	Compute the source term for temperature in the mechanical ventilation system.
HVTOEX	HVTOEX.SOR	Update arrays and assign compartment pressures, temperatures and species to HVAC exterior nodes (to compartments)
HVVIS	HVVIS.SOR	Calculate viscosity of air; used by HVMFLO.
ICOLOR	CF_SET.SOR	Specify color settings in CF_Set.
IDAMAX	BLAS.SOR	Find largest component of double precision vector; BLAS routine.

NAME	DECLARATION	DESCRIPTION
IMOUSE	MOUSE.SOR	Check for the presence of a mouse, save the current mouse vector, and perform initialization.
INDEXI	INDEXI.SOR	Create the index array used by SORTFR.
INIPAR	INIPAR.SOR	Set up labels, titles, units, and so on for graphics display routines in CPlot.
INITAMB	INITAMB.SOR	Initialize the ambient conditions based on TAMB and EAMB.
INITFS	INITFS.SOR	Initialization control routine in CEdit.
INITLOBJ	LIST_OBJ.SOR	Initialize objects arrays for LIST_OBJ.
INITMM	INITMM.SOR	Initialize main memory; required for all modules which will run the CFAST kernel.
INITSLV	INITSLV.SOR	Initialization for solver variables and arrays.
INITSOLN	INITSOLN.SOR	Determine an initial solution to the zone fire modeling equations.
INITSPEC	INITSPEC.SOR	Initialize variables associated with species, based on the ambient conditions.
INITWALL	INITWALL.SOR	Initialize wall variables and arrays, based on conduction settings.
INT2D	CEILHT.SOR	Integrate a function over a region formed by intersecting a rectangle and a circle; used by CEILHT.
INTERP	INTERP.SOR	Interpolate a table of numbers found in two arrays.
INTSQ	CEILHT.SOR	CEILHT integration routine; used by INT2D.
INTTABL	CEILHT.SOR	CEILHT integration routine; used by INT2D.
INTTRI	CEILHT.SOR	CEILHT integration routine; used by INT2D.
IOWAIT>CSPACE	CSPACE.SOR	Pause display for specified number of seconds.
ITHRMF	CF_SET.SOR	Screen in CF_Set to specify paths and default file names.
IUNITS	CF_SET.SOR	Screen in CF_Set to set base and derived units.

NAME	DECLARATION	DESCRIPTION
JAC	JAC.SOR	Evaluate the Jacobian for the solver. (Not used at this time).
KILLPACK	WRITEOT.SOR	Display “FATAL ERROR” message when writing compressed history record in CFAST.
LAYERS	NAILED.SOR	Determine in which layer the person is exposed; used by REPORT.
LENGTH	LENGTH.SOR	Calculate length of CPlot character string.
LENOCO	LENOCO.SOR	Function to calculate length of the main common block (MOCO1A) in storage units.
LISTDEF	LISTDEF.SOR	List CPlot default settings.
LISTOP	LISTOP.SOR	List CPlot available commands.
LIST_OBJ	LIST_OBJ.SOR	List all valid objects in the current objects database.
LIST_TPP	LIST_TPP.SOR	List thermal properties in the current thermo-physical database.
LOADUP	LOADUP.SOR	Load numeric values from graphics descriptor lines into appropriate arrays; used by NPUTG.
LOWERC	LOWERC.SOR	Convert ASCII string to lower case.
LSTFIL	LSTFIL.SOR	Display an ASCII string file name.
LSTINFO	LSTINFO.SOR	Display CFAST version, fire scenario details in CPlot.
LSTOFF>LSTVAL	LSTVAL.SOR	Change the offset number of columns between values in LSTVAL.
LSTRNG	LSTRNG.SOR	Calculate the length of a character string; similar to LENGTH but in CEdit.
LSTVAL	LSTVAL.SOR	Display a row of numeric values.
LUNITS	LUNITS.SOR	Ensure compatible units for variables on a single CPlot graph.
MAKECF	MAKECF.SOR	Write to the configuration file.
MAKTABL	CEILHT.SOR	Set up the integration table used by INTTABL in CEILHT.
MATCH	NAILED.SOR	Verify selected keywords in REPORT.

NAME	DECLARATION	DESCRIPTION
MESS	MESS.SOR	Write a string to IOFILO.
MESSNR>CSPACE	CSPACE.SOR	Similar to MESSNS but writes to current location with current color settings.
MESSNS	MESSNS.SOR	Write to a screen window with specified location and colors.
MODTIM	MODTIM.SOR	Add, delete, or modify points on the CEdit fire timeline.
MVENT	MVENT.SOR	Physical interface routine for the HVAC model.
MVOLAST>MVOUT	MVOUT.SOR	Display mechanical ventilation elevation information.
MVOUT	MVOUT.SOR	Display the mechanical ventilation information from NPUTO in CFAST.
NAILED	NAILED.SOR	Write to IOFILO the tenability information for people nailed in place; used in REPORT.
NEWFILE	NEWFILE.SOR	Get the name of a data file from the CEdit Initialization screen.
NEWTHRM	DISTH1.SOR	Get the name of a new thermal database from the CEdit Thermal Properties screen.
NEW_OBJ	DISOB1.SOR	Get the name of a new objects database from the CEdit Objects screen.
NPUTG	NPUTG.SOR	Read and sort out the graphics descriptors in the input file for display routines in CFAST.
NPUTKB	NPUTKB.SOR	Display prompt and read keyboard response with special keys filtering.
NPUTO	NPUTO.SOR	Write to IOFILO a summary of information read from input file in CFAST.
NPUTOB	NPUTOB.SOR	Read the objects database and initialize arrays for selected objects only in CFAST.
NPUTOBE	NPUTOBE.SOR	Read all objects database names into memory to use for verification in CEdit.
NPUTOR	NPUTOR.SOR	Same as NPUTO but specialized for REPORT.
NPUTP	NPUTP.SOR	Control reading of input file with a post-read verification and cleanup.

NAME	DECLARATION	DESCRIPTION
NPUTQ	NPUTQ.SOR	Read all input file information except the graphics descriptors.
NPUTT	NPUTT.SOR	Read the complete thermophysical properties database into memory for verification.
NPUTTNI	NPUTTNI.SOR	Similar to NPUTT but used by LIST_TPP.
NTRACT	SOLVE.SOR	Key pressed in SOLVE, determine what to do.
NWHCNT	DISVT1.SOR	Recalculate the number of horizontal flow vents for input file in CEdit.
NWOCNT	DISOB1.SOR	Recalculate the number of objects for input file in CEdit.
NWVCNT	DISVT3.SOR	Recalculate the number of vertical flow vents for input file in CEdit.
OBJDFLT	OBJDFLT.SOR	Set up values for “DEFAULT” entry in objects database.
OBJFND	OBJFND.SOR	Search objects database for object name and return pointer to record number.
OBJINT	OBJINT.SOR	Interpolating routine for object fire time history values. Similar to PYROLS.
OBJOUT	OBJOUT.SOR	Write to IOFILO a summary of the objects information from database for specified objects.
OBJSHFL	DISOB1.SOR	Shuffle the ODBNAM and ODBREC arrays if objects are deleted or invalidated.
OBTYPE	OBTYPE.SOR	Determine the type of a specified object.
OFFSET	OFFSET.SOR	Calculate the solver array offsets for each variable.
OPENFILE	OPENFILE.SOR	Open a history file for reading in CPlot, and get the version and creation date.
OPENHELP	OPENHELP.SOR	Open the CEdit help file.
OPENOBJ	OPENOBJ.SOR	Open and read the object database into memory arrays.
OPENSHEL	OPENSHEL.SOR	Read the configuration file, set up the environment, open the output file.

NAME	DECLARATION	DESCRIPTION
OPUT	WRITEOT.SOR	Set up output arrays in PACKOT for ASCII history file.
OUTFIL	OUTFIL.SOR	Display the CEdit Files screen.
OUTI	OUTI.SOR	Display a row of integer data.
OUTPU	OUTPU.SOR	Build the CPlot variable lists of values from the selected file.
OUTPU1	OUTPU1.SOR	Modified form of OUTPU which obtains a single piece of data for CFAST display.
OUTPU16	OUTPU.SOR	Used by OUTPU (not OUTPU1) to return version 1.6 data.
OUTR	OUTR.SOR	Display a row of floating point data.
OUTR6>OUTR	OUTR.SOR	Used by OUTR to write 6 digit floating point data instead of 8.
PACKOT	WRITEOT.SOR	Compression routine for writing the history file.
PALETTE	PALETTE.SOR	Show the palette for the CFAST graphics.
PLOT2	PLOT2.SOR	Reduced version of PLOTITT for the (smaller) CEdit fan curve display.
PLOTBAR>GRAFIT	GRAFIT.SOR	Plot bar at specified location for CFAST graphics.
PLOTCH>GRAFIT	GRAFIT.SOR	Plots alphanumeric characters at specified location for CFAST graphics.
PLOTITG	PLOTITG.SOR	CPlot graphics output.
PLOTITT	PLOTITT.SOR	CEdit text based plots for time dependent variables.
PLOTLN>GRAFIT	GRAFIT.SOR	Plot straight line through specified coordinates in CFAST graphics.
POSIT	POSIT.SOR	Set the type, range, units, etc. for current CEdit field.
POSPAGE>POSIT	POSIT.SOR	Return number of current CEdit field.
PRMCLR	PRMCLR.SOR	Change the CEdit permanent colors.
PRNTLIST	PRNTLIST.SOR	Display CPlot list of variables.
PRNTVALS	PRNTVALS.SOR	Display values in current CPlot list.

NAME	DECLARATION	DESCRIPTION
PRNVAL	PRNVAL.SOR	Simplified version of LSTVAL for use in displaying CEdit thermal database.
PUTLEG	PUTLEG.SOR	Display the legend on a CPlot plot.
PYROLS	PYROLS.SOR	Interpolating routine for specified fire time history values.
QFCLG	CEILHT.SOR	Compute the convective heat transfer flux to the ceiling at location (X,Y)=(Z(1),Z(2)).
QFORM	SNSQE.SOR	Proceed from the computed QR factorization of an M by N matrix A to accumulate the M by M orthogonal matrix Q from its factored form; SNSQE routine.
QRFAC	SNSQE.SOR	Use Householder transformations with column pivoting to compute a QR factorization of the M by N matrix A; SNSQE routine.
QUITFI	QUITFI.SOR	Quit CEdit with check for modified files.
QUSTN	DIOB.SOR	Prompt for option in LIST_OBJ.
R1MPYQ	SNSQE.SOR	Given an M by N matrix A, Compute A*Q where Q is the product of 2*(N - 1) transformations; SNSQE routine.
R1UPDT	SNSQE.SOR	Given an M by N lower trapezoidal matrix S, an M-vector U, and an N-vector V, determine an orthogonal matrix Q such that (S + U*V)*Q is again lower trapezoidal; SNSQE routine.
RAD2	RAD2.SOR	Compute the radiative heat flux for 2 surfaces (extended ceiling and extended floor). Compute the heat absorbed by the lower and upper layers.
RAD4	RAD4.SOR	Compute the radiative heat flux for 4 surfaces (ceiling, upper wall, lower wall, and floor). Compute the heat absorbed by the layers.
RANGE	RANGE.SOR	Display CEdit permitted range in user units.
RDABS	RDABS.SOR	Compute the energy absorbed by the upper and lower layer due to radiation given off by heat emitting rectangles forming the enclosure. Called by RAD2 and RAD4.
RDCNL	RDCNL.SOR	Read a RAPID channel in CPlot.

NAME	DECLARATION	DESCRIPTION
RDFANG	RDFIGSOL.SOR	Define solid angles for fires; called by RAD4.
RDFLUX	RDFLUX.SOR	Calculate the "C" vector in the net radiation equations of Seigel and Howell and the heat absorbed by the lower and upper layer fires due to gas layer emission and fires.
RDFTRAN	RDTRAN.SOR	Define transmission factors for fires; called by RAD4.
RDHEAT	RDHEAT.SOR	Radiation transfer routine between RESID and RAD2 or RAD4.
RDPARFIG	RDFIGSOL.SOR	Calculate the configuration factor between two parallel plates a distance z apart.
RDPRPFIG	RDFIGSOL.SOR	Calculate the configuration factor between two perpendicular plates with a common edge.
RDPRSN	RDPRSN.SOR	Read a person from TENAB file in CPlot.
RDRTRAN	RDTRAN.SOR	Define upper layer transmission factors; called by RAD4.
RDSANG	RDFIGSOL.SOR	Used by RDFANG to define solid angles for fires.
RDSANG1	RDFIGSOL.SOR	Used by RDFANG to define solid angles for fires.
RDTENA	RDTENA.SOR	Read a TENAB channel in CPlot.
READAS	READAS.SOR	Read from IOFILI a single line in ASCII format.
READASTU	READASTU.SOR	Similar to READAS except this contains automatic conversion to upper case. This is to filter commands from the console so they are not case-sensitive.
READBF>READIN	READIN.SOR	Determine if string buffer has been fully parsed, if so, read a new ASCII line.
READCF	READCF.SOR	Read the configuration file for OPENSHEL.
READCV	READCV.SOR	Convert data in graphics descriptor lines to floating point or integer format.
READCV1	READCV1.SOR	Convert CPlot string data.
READFL>READIN	READIN.SOR	Read data from string buffer as an ASCII character string.

NAME	DECLARATION	DESCRIPTION
READHI	READHI.SOR	Read the history file for specified time step.
READIN	READIN.SOR	Parse the line from READAS.
READIT	READIT.SOR	Similar to READIN but for CPlot. ASCII line has not already been read. Positive values only.
READKB	READKB.SOR	Interface for low level keyboard read, mouse, and filter functions.
READMN	READMN.SOR	Similar to READIT but allow a negative sign on data in CPlot.
READOP	READOP.SOR	Read command line options, set up environment, and open designated files.
READPLOT	READPLOT.SOR	Read groupings for CPlot PLOT command.
READRS>READIN	READIN.SOR	Restart START value for string parsing.
REGRESS	REGRESS.SOR	Calculate polynomial coefficients for fan kf by regression analysis.
REPORT	REPORT.SOR	Report generator for existing history file.
REPORTG	REPORTG.SOR	Graphics animation generator for existing history file.
RESID	RESID.SOR	Calculate the right hand side of the predictive equations.
RESTRT	RESTRT.SOR	Called by CFAST to read to specified interval from a history file when RESTRT is specified.
RESULT	RESULT.SOR	Print out the model results of the simulation at the current time to IOFILO.
RESYNC	RESYNC.SOR	Resynchronize the total mass of the species with that of the total mass.
RMOUSE>CSPACE	CSPACE.SOR	Restore mouse vector saved by IMOUSE.
RSLT	RSLT.SOR	Write results from history file to IOFILO in REPORT.
RSMS	MOUSE.SOR	Reset the mouse counters.
RUNFILE	RUNFILE.SOR	Save CEdit input data file using WRITEOF.
RWINDOW>SWINDOW	SWINDOW.SOR	Restore to screen the portion saved in memory by SWINDOW.

NAME	DECLARATION	DESCRIPTION
R_TIME>F_INFO	F_INFO.SOR	CPlot current simulation time.
SAVE	SAVE.SOR	Save an ASCII file in CPlot.
SAVRAP	SAVRAP.SOR	Save a RAPID format file in CPlot.
SCROLLD>CSPACE	CSPACE.SOR	Scroll one line down with ↑ in CEdit.
SCROLLU>CSPACE	CSPACE.SOR	Scroll one line up with ↓ in CEdit.
SETAX	SETAX.SOR	Allow user to set CPlot minimum and maximum values for X and Y axes.
SETAXIN	SETAXIN.SOR	Interface for keyboard read of SETAX values in CPlot.
SETCLR	SETCLR.SOR	Set CEdit default foreground and background colors.
SETLUT	SETLUT.SOR	Setup the color look up table for graphics; Used by DEVICE.LIB.
SETRDAS>READAS	READAS.SOR	Allow sequential use of READAS with direct access files.
SETUNT	SETUNT.SOR	Set conversion factors from user specified units to scientific units.
SET_PAGE>POSIT	POSIT.SOR	Set position to first field on CEdit screen.
SET_UNT>SETUNT	SETUNT.SOR	Set user specified units based on selection from CEdit Units screen.
SLVHELP	SOLVE.SOR	Display function key options available within CFAST run.
SMOUSE>CSPACE	CSPACE.SOR	Save the current mouse vector.
SNSQ	SNSQE.SOR	Find a zero of a system of N nonlinear functions in N variables by a modification of the Powell hybrid method.
SNSQE	SNSQE.SOR	Easy-to-use version of SNSQ.
SOLVE	SOLVE.SOR	Top level of the solver for the predictive equations; Calls RESID, DASSL, etc.
SORTFR	SORTFR.SOR	Sort the room and fire related arrays setup in FIRES by increasing room number order.

NAME	DECLARATION	DESCRIPTION
SQFWST	CEILHT.SOR	Calculate average heat transfer fluxes to lower and upper walls along a vertical line passing through a wall/ceiling jet stagnation point.
SSTRNG	SSTRNG.SOR	Parse a string for space or comma delimited substrings.
SSTRNGP	SSTRNGP.SOR	Similar to SSTRNG but allow parentheses as delimiters in CPlot.
SUNITS	SUNITS.SOR	Display current user selected base and derived units, prompt for changes in CEdit.
SWINDOW	SWINDOW.SOR	Save in memory a portion of the screen to be restored later.
TELLKEYS	TELLKEYS.SOR	Display function key descriptions for current CEdit screen.
TOLOWER	TOUPPER.SOR	Convert ASCII string to lower case.
TOPSCR	TOPSCR.SOR	Clear the screen and set foreground and background colors.
TOUPPER	TOUPPER.SOR	Convert ASCII string to upper case.
TOXIC	TOXIC.SOR	Calculate the parts per million of gases and the concentration time dose.
TOXICB	TOXICB.SOR	Setup and display the labels for table of doses, etc. in CFAST.
TOXICH	TOXICH.SOR	Setup and display the table for doses, etc. in CFAST.
TOXICR	TOXICR.SOR	Setup and display the values for table of doses, etc. in CFAST.
TURNKBON>NPUTKB	NPUTKB.SOR	Turn keyboard editing on to allow polling.
UNPACK	DREADIN.SOR	Decompression routine for reading the history file.
UP_FIRE	DISFIR.SOR	Calculate third value from last two specified of heat of combustion, pyrolysis rate, and rate of heat release on CEdit Fire screen.
UP_VENT	DISVT1.SOR	Cleanup routine on CEdit Vents(doors,...) screen to verify that flows are specified the same in both directions.

NAME	DECLARATION	DESCRIPTION
VDCO2	NAILED.SOR	Multiplication factor for the effect on ventilation allowing for the effect of the increased RMV caused by carbon dioxide on the rate of uptake of other toxic gases.
VENT	VENT.SOR	Calculation of the flow of mass, enthalpy, oxygen and other products of combustion through a vertical, constant-width vent in a wall segment common to two compartments.
VENTCF	VENTCF.SOR	Calculate the flow of mass, enthalpy, and products of combustion through a horizontal flow vent joining an upper space 1 to a lower space 2.
VFLOW	VFLOW.SOR	Interface between CFAST and the vertical flow vent physical routines.
VWPRT	VWPRT.SOR	Calculate the window size for a view within a small window.
WAIT100>CSPACE	CSPACE.SOR	Wait specified time for response from mouse; kept for backwards compatibility.
WDDRAW	WDDRAW.SOR	Draw the CPlot stroke characters.
WHICHONE	WHICHONE.SOR	CPlot command processor.
WINDOW>CSPACE	CSPACE.SOR	Create window with specified attributes and positions.
WRITEOF	WRITEOFE.SOR	Write CEdit specified data to the data file.
WRITEOT	WRITEOT.SOR	Write a record (binary) of a history file. See DREADIN.
WRT_GEO	WRITEOFE.SOR	Write geometry section record to input file.
WRT_THP	WRITEOFE.SOR	Write thermal property record to input file.
WR_COEF	DISFAN.SOR	Display fan coefficients on Fans subpage of CEdit Fans, Ducts,... screen.
WR_CONN	DISVT2.SOR	Display line of HVAC external connections on CEdit Fans, Ducts,... screen.
WR_DUCT	DISDCT.SOR	Display line of HVAC internal connections on CEdit Ducts subpage.
WR_FAN	DISFAN.SOR	Display line of HVAC fans on CEdit Fans subpage of Fans, Ducts,... screen.

NAME	DECLARATION	DESCRIPTION
WR_FLOW	DISFAN.SOR	Display flow values on Fans subpage of CEdit Fans, Ducts,... screen.
WR_OBJ	DISOB1.SOR	Display information for one object on CEdit Objects screen.
WR_VENT	DISVT1.SOR	Display information for one horizontal flow vent on CEdit Vents(doors,...) screen.
WR_VVNT	DISVT3.SOR	Display information for one vertical flow vent on CEdit Vents(ceiling,...) screen.
WSET	WSET.SOR	Set Initial wall variables and arrays for one surface; used by INITWALL.
XERROR	AAUX.SOR	Process an error (diagnostic) message in SOLVE.
XERRWV	AAUX.SOR	A simplified version of XERROR.
XFDCO	NAILED.SOR	Fractional effective dose due to CO for specified interval in REPORT; see FDCO.
XFDHCN	NAILED.SOR	Fractional effective dose due to HCN in specified interval in REPORT; see FDHCN.
XFDO2	NAILED.SOR	Fractional effective dose due to O2 in specified interval in REPORT; see FDO2.

The following block data statements are not executed by any of the modules, but provide initialization of the named common blocks. They must be included in the appropriate makefile.

Block Data Name	Source File	Common block initialized
INITBK	INITBK.SOR	CFIN not included in INITLV, INITUN
INITCF	INITCF.SOR	CRDATE variable in MOCO1A
INITCS	READCF.SOR	CSHELL version, files, and device numbers.
INITDI	INITDI.SOR	MOCO2A and DFLTS
INITLV	INITLV.SOR	Labels in CFIN.
INITPL	INITPL.SOR	DFLTS, DFLTS1, PLTFLT for plotting and defaults.
INITSLVB	INITSLV.SOR	Solver variables.
INITUN	INITUN.SOR	Unit conversion arrays in CFIN.

Block Data Name	Source File	Common block initialized
NAILEDF	NAILED.SOR	NAILED

Appendix C: CFAST Common Block Variables

Name	Type	Common Block	Description
AA(NR,NR,4)	R*8	/MOCO1A/	Flow from lower layer to lower layer (kg/s).
ACON(MXORM,MXPRD)	R*8	/PRODS/	Concentration of species in mechanical ventilation ducts. **No longer used.**
ACTIVS(NS)	L*4	/MOCO1A/	Logical switch to tell which species are active (interacts with "allowed").
ADUMPF	CHAR*60	/CONFG2/	Name of the file used for an ASCII dump in DISPLAYC.
ADVFEA	L*4	/CONFG1/	Logical switch for availability of advanced features in the data editor, such as ability to run the model from the data editor, set colors, ...
AFIRED(NV)	R*8	/MOCO1A/	Time history for area of fire (m^2).
AHISTI	I*4	/MOCO2A/	Counter indicating AHISTOV element for ASCII dump data in DISPLAYC.
AHISTOV(25)	R*4	/MOCO2A/	Stores ASCII dump data in DISPLAYC.
AHVPTOL	R*8	/SOLVPRM/	Absolute error tolerance for pressure at interior HVAC nodes.
AHVTTOL	R*8	/SOLVPRM/	Absolute error tolerance for temperature in HVAC ducts.
AIRAK	R*8	/PHYCNS/	Short pool/wall ratio (FSM - not implemented)
ALGTOL	R*8	/SOLVPRM/	Tolerance for algebraic solver.
ALLOWED(NS)	L*4	/BLANK-COMMON/	Which species can be set relative to the fuel
ANGLE	R*4	/GRFTYP/	Angle of a label written by LABEL in DEVICE.LIB. Pulled in by GRAFIT.SOR and WDDRAW.SOR.
AO(MFT)	R*8	/MOCO1A/	Area of HVAC simple fitting IFT (m^2).
APS(NR)	R*8	/MOCO1A/	Current area of the specified fire (m^2). **(Not currently used).**
APSPM	R*8	/COMINT/	**No longer used.**
APTOL	R*8	/SOLVPRM/	Absolute error tolerance for compartment pressures.
AR(NR)	R*8	/MOCO1A/	Floor area of a compartment (in current version area of ceiling = area of floor).

Name	Type	Common Block	Description
AREXT(MEXT)	R*8	/MOCO1A/	Cross-sectional area of the opening from a compartment to a mechanical ventilation duct (m_2).
AS(NR,NR,4)	R*8	/MOCO1A/	Flow from lower to upper layer (kg/s).
ASL(NR,NR,4)	R*8	/MOCO1A/	Entrainment from upper into lower layer (kg/s).
ATOL	R*8	/SOLVPRM/	Absolute error tolerance for compartment layer volume and temperatures.
AWTOL	R*8	/SOLVPRM/	Absolute error tolerance for wall surface temperature equation.
BAR(6,5)	R*4	/MOCO2A/	XYZ coordinate pairs for up to 5 bar charts.
BART(16,2)	CHAR*50	/MOCO2B/	Titles for bar charts.
BFIRED(NV)	R*8	/MOCO1A/	Burning rate time history (kg/s).
BFLO(MBR)	R*8	/MOCO1A/	Mass flow rate through HVAC branch IB (kg/s).
BR(NR)	R*8	/MOCO1A/	Breadth of a compartment (m).
BW(NR,NR,4)	R*8	/MOCO1A/	Width of vent (m) (modified by QCVENT).
C1	R*8	/AINTCH/	Intermediate value for the ceiling jet calculation.
C2	R*8	/AINTCH/	Intermediate value for the ceiling jet calculation.
C3	R*8	/AINTCH/	Intermediate value for the ceiling jet calculation.
CARDTYPE	CHAR*4	/CONFG2/	Type of graphics adaptor.
CBUF	CHAR*256	/INPUT1/	Character buffer used for editing (CEdit,...)
CCO2(NV)	R*8	/MOCO1A/	Net carbon production rate (fraction relative to CO^2).
CE(MBR)	R*8	/MOCO1A/	Conductance of HVAC branch IB.
CF_EXIST	L*4	/CONFG1/	Logical, set to true if the named configuration file exists.
CHSIZE(9)	R*4	/GRFTYP/	Array of parameters in DEVICE.LIB indicating the size and placement of characters in CFAST graphics. Pulled in by GRAFIT.SOR and WDDRAW.SOR.
CHV(MBR)	R*8	/BLANK-COMMON/	Convective heat transfer coefficient for HVAC duct IB.

Name	Type	Common Block	Description
CJETON(NWAL+1)	L*4	/MOCO1A/	Array for ceiling jet settings. Element 1-ceiling, 3,4-wall, 5-set if 1, 3, or 4 are set, (2 is not used).
CNAME(NWAL,NR)	CHAR*8	/MOCO1E/	Database thermal property material name specified for surface in input file.
CNUM(16,25)	I*4	/MOCO2A/	Compartment number for each of 16 variables up to 25 instances in CPlot.
CNVG(5)	R*8	/MOCO1A/	Configuration and tolerance values for the numerical solver. **No longer used.**
CO(MFT)	R*8	/MOCO1A/	Flow coefficient for HVAC simple fitting IFT.
COCO2(NV)	R*8	/MOCO1A/	Relative CO/CO_2 production rate.
COFSET(NVAR)	I*4	/PLTPAR/	Compartment offset for connections. **No longer used.**
CONEF(5)	CHAR*60	/OBJCT2/	Cone calorimeter file names for an entry in objects database.
CONEN	I*4	/OBJCT3/	Number of cone calorimeter files for an entry in objects database.
CONFIG	CHAR*60	/CONFG2/	Name of the configuration file.
COUNT	I*4	/READ1C/	Number of characters in INBUF string remaining to be parsed.
CP	R*8	/MOCO1A/	Heat capacity of air at constant pressure (J/kg/K).
CQ	R*8	/PHYCNS/	First time stretching parameter (FSM - not implemented).
CRDATE(3)	I*4	/MOCO1A/	Creation date of the model (day, month and year).
CSOPEN	L*4	/CONFG1/	**Set but not used.**
CTIME	R*8	/PHYCNS/	FSM - not implemented.
CURRENT	CHAR*64	/CONFG2/	Current path, which can be different from the database and data paths. **Set but not used.**
CW(MXSLB,NWAL,NR)	R*8	/MOCO1A/	Specific heat of thermal material for each specified surface in input file (see LCW).
DA(MDT)	R*8	/MOCO1A/	Area of HVAC duct id (m^2).
DASSLFTS	R*8	/SOLVPRM/	Initial step size used by the solver, DASSL. Default value is .005 (s).
DE(MDT)	R*8	/MOCO1A/	Effective diameter of HVAC duct id (m).

Name	Type	Common Block	Description
DEBUG(MXOPT)	I*4	/DEBUG/	**Not yet implemented.**
DELTAT	R*8	/MOCO1A/	Time step used by the model.
DELTE	R*8	/COMINT/	**No longer used.**
DENAIR	R*8	/PHYCNS/	Density of air (kg/m^3). (FSM - not implemented)
DFILE	CHAR*60	/CONFG2/	Name of the data file read from the command line or the configuration file.
DFMAX(MFAN)	R*8	/MOCO1A/	Derivative of HVAC fan curve at HMAX(k).
DFMIN(MFAN)	R*8	/MOCO1A/	Derivative of HVAC fan curve at HMIN(k).
DHVPRSYS(MNODE,NS)	R*8	/BLANK-COMMON/	Time rate of change of product IPROD associated with HVAC system ISYS.
DIRS12(10)	I*4	/VNTSLB/	Direction of flow in slab i for natural flow through a vent.
DL(MDT)	R*8	/MOCO1A/	Length of HVAC duct id (m).
DLEN	I*4	/CONFG1/	Number of characters in the database path (DPATH); zero implies not used.
DPATH	CHAR*64	/CONFG2/	Path for executables and databases.
DPV1M2(10)	R*8	/VNTSLB/	Pressure drop for slab i for natural flow through a vent.
DPZ(MNODE,MCON)	R*8	/MOCO1A/	Hydrostatic pressure difference between node i and node k.
DQ	R*8	/PHYCNS/	Second time stretching parameter (FSM - not implemented).
DR(NR)	R*8	/MOCO1A/	Depth of a compartment (m).
DT	R*8	/DERVS/	The time step set by DASSL.
DTIME	R*8	/PHYCNS/	FSM - not implemented.
DUCTAR(MDT)	R*8	/MOCO1A/	Absolute roughness of HVAC duct walls.
DUCTCV	R*8	/BLANK-COMMON/	Convection coefficient for HVAC ducts.
DUMPF	CHAR*60	/CONFG2/	History file - set by NPUTQ.
EDG(2,NEMX)	I*4	/MOCO2A/	Vertices for edge NEMX in a graphics or BUILD picture file.
EDGS(4,NEMX)	I*4	/MOCO2A/	Attributes of edge NEMX in a graphics or BUILD picture file.

Name	Type	Common Block	Description
EME(NR)	R*8	/MOCO1A/	Plume entrainment rate (kg/s).
EMISIV	R*8	/PHYCNS/	Panel surface emissivity (FSM - not implemented).
EMP(NR)	R*8	/MOCO1A/	Pyrolysis rate of the fire source (kg/s).
EMS(NR)	R*8	/MOCO1A/	Plume flow rate into the upper layer (kg/s).
EPA(NR)	R*8	/MOCO1A/	External ambient pressure.
EPW(NWAL,NR)	R*8	/MOCO1A/	Emissivity of the interior wall surface (dimensionless).
ERA(NR)	R*8	/MOCO1A/	External ambient mass density (kg/m^3).
ETA(NR)	R*8	/MOCO1A/	External ambient temperature (K).
ETIME	R*8	/COMINT/	**No longer used.**
EXPA	R*8	/BLANK-COMMON/	External ambient pressure at the reference elevation.
EXRA	R*8	/BLANK-COMMON/	External ambient density at the reference elevation (kg/m^3). **No longer used.**
EXSAL	R*8	/BLANK-COMMON/	Reference elevation of the outside of the structure.
EXSET	L*4	/BLANK-COMMON/	Logical variable set if the external ambient is set separately from the internal ambient.
EXTA	R*8	/BLANK-COMMON/	External ambient temperature at the reference elevation (K).
FACTRD(MXUNT)	R*4	/INPUT2/	Conversion factors to convert from user specified units to scientific units for each of the MXUNT derived quantities such as volume, density, specific heat, etc. See CUNITS for list of derived quantities.
FACTRU(MXUNT)	R*4	/INPUT2/	Conversion factors to convert from scientific units to user specified units for each of the MXUNT derived quantities such as volume, density, specific heat, etc. See CUNITS for list of derived quantities. Used by CEdit for display.
FASTFOR	I*4	/INPUT2/	**No longer used.**
FIRST	I*4	/READ1C/	First position within INBUF string of a space delimited substring (see LAST, START).

Name	Type	Common Block	Description
FKW(MXSLB,NWAL,NR)	R*8	/MOCO1A/	Thermal conductivity of material slab for specified surface in input file (see LFKW).
FLMTMP	R*8	/PHYCNS/	Thermal radiation temperature of flames (FSM - not implemented).
FLW(MXSLB,NWAL,NR)	R*8	/MOCO1A/	Thickness of a slab (m) for specified surface in input file (see LFLW).
FOOTLT	R*8	/PHYCNS/	Effective flame foot length (FSM - not implemented).
FPOS(3)	R*8	/MOCO1A/	Position of the main fire. 1-depth(X), 2-breadth(Y), 3-height(Z).
FSAVG	R*8	/PMWRT1/	**No longer used.**
G	R*8	/MOCO1A/	Gravitational constant (9.806 m/s).
GAMMA	R*8	/MOCO1A/	cp/cv for air (1.4).
GASCNT(16)	I*4	/MOCO2A/	Count of the number of instances of each of the 16 plot variables.
GASN(2,16,25)	I*4	/MOCO2A/	From/to pointer for vents in DISPLAYC.
GASPTR(5,16,25)	I*4	/MOCO2A/	Cross-reference pointer for plot variable - 5 plot types, 16 variables, 25 instances per variable.
GFILE	CHAR*60	/CONFG2/	Geometry file (not used in this version) intended for compartment interconnections.
GMWF	R*8	/MOCO1A/	Gram molecular weight (g), default (16).
GRAPH(6,5)	R*4	/MOCO2A/	X,Y,Z positions of up to 5 graphs.
GRCOLR(5)	I*4	/MOCO2A/	Axis and frame color of each of the 5 graphs.
GRFT(16,2)	CHAR*50	/MOCO2B/	X,Y axis labels for each of the 16 variables.
GTRAIL	I*4	/INPUT2/	Flag indicating graphics descriptors to be written to input file by CEdit. (2 - CEdit standard graphics, otherwise user specified graphics previously in file).
H	R*8	/AINTCH/	Intermediate variable for ceiling jet.
HCLBF(7,NWAL,NR)	R*8	/MOCO1A/	“B” constants used in HCl deposition calculation for specified surface in input file (see LHCLBF).
HCLDEP	I*4	/MOCO1A/	HCl deposition switch on/off.

Name	Type	Common Block	Description
HCLF(NV)	R*8	/MOCO1A/	Time history fuel array for HCl.
HCNF(NV)	R*8	/MOCO1A/	Time history fuel array for HCN.
HCOMBA	R*8	/MOCO1A/	Heat of combustion, scalar value, not time specific. Initialization only from CHEMI keyword line in input file.
HCPAIR	R*8	/PHYCNS/	Solid/visible flame height ratio (FSM - not implemented).
HCRATIO(NV)	R*8	/MOCO1A/	Time history of the hydrogen/carbon ratio of the fuel.
HCRATT	R*8	/BLANK-COMMON/	Current hydrogen/carbon ratio in the fuel for a specified fire (dimensionless).
HEADER	L*4	/CONFG1/	Logical used to determine if disclaimer is to be displayed. On by default.
HEATLP(NR)	R*8	/MOCO1A/	Heat release rate in the plume in the lower layer (W).
HEATUP(NR)	R*8	/MOCO1A/	Heat release rate in the plume in the upper layer (W).
HEATVF(NR)	R*8	/MOCO1A/	Heat release in a vent (sum of all vents between compartments).
HFIRED(NV)	R*8	/MOCO1A/	Time history for height of the base of the fire (m).
HFLR(NR)	R*8	/MOCO1A/	Absolute height of the floor of a compartment (m).
HFOTOP	R*8	/COMINT/	**No longer used.**
HFTOLD	R*8	/COMINT/	**No longer used.**
HH(NR,NR,4)	R*8	/MOCO1A/	Height of the vent soffit (top) relative to the floor (m).
HHP(NR,NR,4)	R*8	/MOCO1A/	Absolute height of the vent soffit (top) (m).
HL(NR,NR,4)	R*8	/MOCO1A/	Height of the vent sill (bottom) relative to the floor (m).
HLP(NR,NR,4)	R*8	/MOCO1A/	Absolute height of the vent sill (bottom) (m)
HMAX(MFAN)	R*8	/MOCO1A/	Maximum head pressure value for HVAC fan (Pa)
HMFLOW(NR,2)	R*8	/BLANK-COMMON/	**No longer used.**
HMFNET(2,NR,NR)	R*8	/MOCO1A/	**No longer used.**
HMIN(MFAN)	R*8	/MOCO1A/	Minimum head pressure value for HVAC fan (Pa)

Name	Type	Common Block	Description
HOCBMB(NV)	R*8	/MOCO1A/	Time history for the heat of combustion of a specified fire (J/kg). Calculated from BFIRED and QFIRED if the FMASS and FQDOT keywords are specified.
HR(NR)	R*8	/MOCO1A/	Interior height of a compartment (m).
HRL(NR)	R*8	/MOCO1A/	Absolute height of the floor of a compartment (m).
HRP(NR)	R*8	/MOCO1A/	Absolute height of the ceiling of a compartment (m).
HTCT	R*8	/AINTCH/	Intermediate value for the ceiling jet.
HTFLOW(NR,2)	R*8	/BLANK-COMMON/	**No longer used.**
HTFNET(2,NR,NR)	R*8	/BLANK-COMMON/	**No longer used.**
HTOT(NR)	R*8	/BLANK-COMMON/	**No longer used.**
HVBCO(MFAN,MFCOE)	R*8	/MOCO1A/	Coefficients of HVAC fan curve polynomial.
HVCONC(MBR,NS)	R*8	/MOCO1A/	HVCONC(i,j) is the concentration of species j in branch i of an HVAC system.
HVDARA(MBR)	R*8	/BLANK-COMMON/	Surface area of HVAC duct IB.
HVDELT	R*8	/MOCO1A/	**No longer used.**
HVDVOL(MBR)	R*8	/MOCO1A/	Volume of HVAC branch IB.
HVEFLO(2,MEXT)	R*8	/MOCO1A/	Mass flow into upper <lower> layer of compartment connected to HVAC external node IB.
HVELXT(MEXT)	R*8	/MOCO1A/	Elevation of HVAC exterior nodes relative to compartment floor (m).
HVEXCN(MEXT,NS)	R*8	/MOCO1A/	Species concentration at HVAC external nodes (kg/m^3).
HVEXTT(MEXT)	R*8	/MOCO1A/	Temperature of flow through HVAC external node i.
HVFLOW(MNODE,MCON)	R*8	/MOCO1A/	HVFLOW(i,j) is the mass flow rate to node i from the jth node to which it is connected.
HVFRAC(2,MEXT)	R*8	/BLANK-COMMON/	Fraction of HVAC external node i connected to upper <lower> layer of compartment associated with the external node.
HVGHT(MNODE)	R*8	/MOCO1A/	Elevation of HVAC node i relative to reference elevation.

Name	Type	Common Block	Description
HVGRAV	R*8	/MOCO1A/	Acceleration of gravity, 9.8 (m/s^2).
HVMFSYS(MXHVSYS)	R*8	/BLANK-COMMON/	Mass flow rate into HVAC system ISYS.
HVNODE(2,MEXT)	I*4	/MOCO1A/	Mapping in HVAC. HVNODE(1,j)-compartment connected to exterior node j, HVNODE(2,j)-node corresponding to exterior node j.
HVORIEN(MEXT)	I*4	/MOCO1A/	Orientation of an external HVAC connection (1 - horizontal, 2 - vertical).
HVP(MNODE)	R*8	/MOCO1A/	Relative pressure at HVAC node i.
HVRGAS	R*8	/MOCO1A/	Ideal gas law constant.
HVT	R*8	/BLANK-COMMON/	**No longer used.**
HVTM(MXHVSYS)	R*8	/BLANK-COMMON/	Total mass in HVAC system ISYS.
HWJ(NWAL,NR)	R*8	/MOCO1A/	HCl density on the wall (g/m^2). Initialized to 0). **No longer used.**
HWJDOT(NWAL,NR)	R*8	/BLANK-COMMON/	Rate of deposition of hydrogen chloride to a wall surface ($kg/s/m^2$). **No longer used.**
HZRD	R*4	/MOCO2A/	**No longer used.**
HZRDPL(4,16)	R*4	/MOCO2A/	HZRDPL(i,j) is the maximum allowed value assigned to the hazard level i for the type of hazard j. Hazard level 1-low to 4-dead. For example, type 6 is interface height.
IBRD(MDT)	R*8	/MOCO1A/	Pointer to HVAC resistive branch with duct id
IBRF(MFT)	I*4	/MOCO1A/	Pointer to HVAC resistive branch with fitting IFT.
ICBG	I*4	/CONFG1/	Background color for normal text in CEdit and CF_Set.
ICEBG	I*4	/CONFG1/	Background color for errors in CEdit, etc.
ICEMS	I*4	/CONFG1/	Foreground color for errors in CEdit, etc.
ICHBG	I*4	/CONFG1/	Background color for help in CEdit, etc.
ICHDR	I*4	/CONFG1/	Foreground color for headers in CEdit, etc.
ICHLP	I*4	/CONFG1/	Foreground color for help in CEdit, etc.

Name	Type	Common Block	Description
ICHRS	I*4	/DFLTS/	Color variable referenced in CPlot but **does not seem to be used**
ICMBG	I*4	/CONFG1/	Background color for messages in CEdit and CF_Set.
ICMSG	I*4	/CONFG1/	Foreground color for messages in CEdit and CF_Set.
ICMV(MNODE,MCON)	I*4	/MOCO1A/	ICMV(i,j) is the HVAC branch j connected to node i.
ICOLOL(34)	I*4	/CONFG1/	Array of integer values read from PASSFILE. Equivalenced to ICHDR in READOP.
ICPRO	I*4	/CONFG1/	Foreground color for protected text in CEdit.
ICSUB	I*4	/CONFG1/	Foreground color for subheadings in CEdit.
ICTXT	I*4	/CONFG1/	Foreground color for normal text in CEdit and CF_Set.
IDEF	I*4	/DFLTS/	Default compartment number for variable selection in CPlot.
IDEVC	I*4	/DFLTS/	The device number specified by the DEVICE line in the input file for CFAST graphics.
IDEVIC	I*4	/DEVTYP/	The device number used within the DEVICE.LIB, pulled into CFAST by GRAFIT.SOR.
IDEVO	I*4	/DFLTS/	Device number of printed output in CPlot.
IDIAG	I*4	/MOCO1A/	**No longer used.**
IFIRED	I*4	/MOCO1A/	Current interpolation time for specified fire - integer pointer. **No longer used.**
IHIST	I*4	/PHYCNS/	Scaled time parameter (FSM - not implemented)
IHMLAR(2,NR,NR)	I*4	/BLANK-COMMON/	**No longer used.**
IN(MNODE,MCON)	I*4	/MOCO1A/	IN(i,j) is the pointer to HVAC node j connected to node i.
INBUF	CHAR*256	/READ2C/	Character string buffer used by file input and parsing routines.
INDEX(16)	I*4	/MOCO2A/	Pointer into INIPAR list of variables.
INPUT(4096)	I*4	/MOCO1A/	Equivalenced with GAMMA in RESTRT.

Name	Type	Common Block	Description
INTER(NR)	R*8	/BLANK-COMMON/	Equivalenced with QFR in NPUTQ.
IOFILI	I*4	/CONFG1/	Input file unit number - defined in the INITCS block data.
IOFILO	I*4	/CONFG1/	Output file unit number - defined in the INITCS block data.
IOUTF	I*4	/VNTSLB/	**No longer used.**
IPAR2(2)	I*4	/SOLVPRM/	Parameter used by DASSL. 1-number of equations solved, 2-flag indicating whether all residuals (including species) or only those residuals used by DASSL are to be calculated.
IPLOT	I*4	/PHYCNS/	FSM - not implemented.
IRMTRM	I*4	/PHYCNS/	Mode parameter of thermal radiation solution (FSM - not implemented).
IRVRSE	I*4	/GRFTYP/	Pulled in by GRAFIT.SOR and WDDRAW.SOR from DEVICE.LIB.
ISPDEF	I*4	/DFLTS/	Default species type for variable selection in CPlot.
ISTOCH	I*4	/PHYCNS/	Heat release scaling factor (FSM - not implemented).
ITKWIT	I*4	/GRFTYP/	Pulled in by GRAFIT.SOR and WDDRAW.SOR from DEVICE.LIB.
ITMMAX	I*4	/MOCO1A/	Maximum number of time steps (#).
ITMSTP	I*4	/MOCO1A/	Current time step (#).
ITODEF	I*4	/DFLTS/	Default “to” compartment for plotting vent flow.
IVERS	I*4	/MOCO1A/	Current version (major version•100 plus “sub” version number).
IVVENT(MXVENT,2)	I*4	/VENTS/	Compartment numbers on either side of vent i.
IWBOUND	I*4	/WNODES/	Type of exterior wall boundary condition. 1-constant temperature, 2-insulated, 3-flux (calculated assuming ambient outside condition).
IWRITE	I*4	/PMWRT1/	**No longer used.**
IX	I*4	/READ1C/	Integer numeric entry from keyboard.
IZHVBSYS(MBR)	I*4	/BLANK-COMMON/	System number associated with HVAC branch IB.

Name	Type	Common Block	Description
IZHVIE(MNODE)	I*4	/BLANK-COMMON/	External node corresponding to HVAC node i.
IZHVMAPE(MNODE)	I*4	/BLANK-COMMON/	Inverse of IZHVMAPI. Given the HVAC node i, IZHVMAPE(i) is the corresponding interior node DASSL solves for. If IZHVMAPE(i) = 0, the HVAC node i is not determined by the solver.
IZHVMAPI(MNODE)	I*4	/BLANK-COMMON/	Inverse of IZHVMAPE. Given the DASSL interior node i, IZHVMAPI(i) is the corresponding HVAC node number.
IZHVSYS(MNODE)	I*4	/BLANK-COMMON/	System number associated with HVAC node i.
IZPMAP(NS+2)	I*4	/CENVIRO/	Some species data structures do not leave space for inactive species. IZPMAP provides a mapping between packed and unpacked species arrays.
IZVENT(MXVENT,5)	I*4	/VENTS/	The first and second compartments containing vent i. IZVENT(i,3) is a vent number used by obsolete data structures.
KBCNT	I*4	/MSKBD/	Number of characters currently in keyboard buffer. Used by NPUTKB and READKB.
KBFRST	I*4	/MSKBD/	First element number in the keyboard buffer to be displayed on the input line. Used by NPUTKB and READKB.
KBINS	L*4	/MSKBD/	**No longer used.**
KBLAST	I*4	/MSKBD/	Last element number in the keyboard buffer to be displayed on the input line. Used by NPUTKB and READKB.
KBLNGT	I*4	/MSKBD/	Longest substring which can be displayed on the input line. Currently 19. Used by NPUTKB and READKB.
KBPNT	I*4	/MSKBD/	Screen column number in which KBLAST element is displayed. Used by NPUTKB and READKB.
KBSET	L*4	/MSKBD/	Logical indicating existence of characters in keyboard buffer to be processed. Used by NPUTKB and READKB.
KBSTRT	I*4	/MSKBD/	Screen column number in which KBFRST element is displayed. Used by NPUTKB and READKB.
KBXBF	I*4	/MSKBD/	Longest string allowed for this input. Used by NPUTKB and READKB.

Name	Type	Common Block	Description
KEYWRD(NVAR)	CHAR*12	/PLTPAR/	Array of keywords available for plots.
LABANGL(2,5)	R*4	/MOCO2A/	Angle of the 5 available graphics labels.
LABCOLR(5)	I*4	/MOCO2A/	Color of the 5 available graphics labels.
LABEL(NVAR)	CHAR*40	/PLTPAR/	Y axis labels for graphs in CPlot.
LABLE(6,5)	R*4	/MOCO2A/	X,Y,Z positions of the 5 graphics labels.
LABTEXT(5)	CHAR*50	/MOCO2B/	Text for the 5 CFAST graphics labels.
LABUL(NVAR)	CHAR*40	/PLTPAR/	Equivalence to LABEL(NVAR). Necessary in DISPLAYC.SOR due to routine named LABEL.
LAST	I*4	/READ1C/	Last position within INBUF string of a space delimited substring. (see FIRST, START)
LAYDEF	I*4	/DFLTS/	Default layer for plotting.
LBACK	I*4	/INPUT2/	Default background color for screen. Used by CEdit and CF_Set.
LBUF	CHAR*256	/INPUT1/	Temporary string buffer used for path construction and command line parsing.
LCCOL	I*4	/INPUT2/	Previously used to track last column accessed on a subpage of the Geometry screen in CEdit. **No longer used.**
LCOPY	I*4	/MOCO1A/	Number of “hard” copies for each graphics output (used for movies).
LCROW	I*4	/INPUT2/	Previously used to track last row accessed on a subpage of the Geometry screen in CEdit. **No longer used.**
LCW(MXSLB,NTHMX)	R*8	/MOCO1D/	Specific heat of each thermal material from thermal database (see CW).
LDIAGO	I*4	/MOCO1A/	History interval (#).
LDIAGP	I*4	/MOCO1A/	Display (graphics) interval (#).
LEGEND(6)	R*4	/MOCO2A/	**No longer used.**
LEGENDON	L*4	/MOCO2A/	**No longer used.**
LEPW(NTHMX)	R*8	/MOCO1D/	Local emissivity
LESPEC(2)	I*4	/MOCO2A/	**No longer used.**
LFBO	I*4	/MOCO1A/	Compartment of origin (1 to NR-1)
LFBT	I*4	/MOCO1A/	Type of fire (1, 2, ...)

Name	Type	Common Block	Description
LFKW(MXSLB,NTHMX)	R*8	/MOCO1D/	Thermal conductivity of a material slab for each entry in thermal database (see FKW).
LFLW(MXSLB,NTHMX)	R*8	/MOCO1D/	Thickness of a slab (m) for each entry in thermal database (see FLW).
LFMAX	I*4	/MOCO1A/	Number of points in a fire specification including initial time.
LFORE	I*4	/INPUT2/	Default foreground color for screen. Used by Cedit and CF_Set.
LHCLBF(7,NTHMX)	R*8	/MOCO1D/	"B" constants for Hcl deposition calculation for each entry in thermal database (see HCLBF).
LIMO2	R*8	/MOCO1A/	Limiting oxygen index as a percent (default is 10 percent).
LNSLB(NTHMX)	I*4	/MOCO1D/	Number of slabs for each entry from thermal database (see NSLB).
LOFSET(NVAR)	I*4	/PLTPAR/	**No longer used.**
LOGERR	I*4	/CONFG1/	Unit number for log file. Zero implies log file is not used.
LOGOON	L*4	/MOCO2A/	**No longer used.**
LOGTIM	CHAR*11	/INPUT1/	Current time for log file output.
LPRINT	I*4	/MOCO1A/	Print interval.
LRON	L*4	/INPUT2/	Enable keyboard entry.
LRW(MXSLB,NTHMX)	R*8	/MOCO1D/	Material density of a boundary slab (kg/m^3) for each entry from thermal database (see RW)
LSW	I*4	/DEVTYP/	Pulled in from DEVICE.LIB by GRAFIT.SOR.
LTSW	I*4	/DEVTYP/	Pulled in from DEVICE.LIB by GRAFIT.SOR.
LUDIAG	I*4	/GRFTYP/	Logical unit for diagnostic writes from graphics software in DEVICE.LIB. Pulled in by GRAFIT.SOR and WDDRAW.SOR.
LUHSET	I*4	/GRFTYP/	The unit number used by DEVICE.LIB to read in Hershey set characters. Pulled in by GRAFIT.SOR and WDDRAW.SOR.
LUOUT	I*4	/DEVTYP/	The unit number used by DEVICE.LIB to write graphics to disk. Pulled in by GRAFIT.SOR.

Name	Type	Common Block	Description
MAPLTW(NWAL)	I*4	/BLANK-COMMON/	Mapping for the convective and radiative phenomena. **No longer used.**
MASS(2,NR,NS)	R*8	/MOCO1A/	Mass in a compartment layer of species i (1 to NS).
MAXCT	I*4	/MOCO1D/	Number of entries in the thermal database (maximum is 57 now).
MAXIND(6)	I*4	/MOCO2A/	Count of the number of occurences of each output type in DISPLAYC.
MAXINR	I*4	/MOCO1A/	Maximum size of inner time step for time splitting (only in CEdit, will be obsolete)
MAXSZ(NVAR)	I*4	/PLTPAR/	Maximum number of characters used to specify a variable input in CPlot.
MAXTIT	I*4	/INPUT2/	Longest title display allowed in CEdit and CF_Set. Effects menu bar and screen headings.
MAXUNT(MXKYWD)	I*4	/INPUT2/	Maximum number of characters to accurately specify one of the MXKYWD units available on the Units page of Cedit. See INITUN for list of available units.
MDELAY	I*4	/MSKBD/	Number of clock ticks which signifies a double mouse click.
MFIRET(NS)	R*8	/MOCO1A/	Mass release rate of species i (transient).
MINMAS	R*8	/MOCO1A/	Minimum allowed value in array MASS.
MINSZ(NVAR)	I*4	/PLTPAR/	Minimum number of characters used to specify a variable input in CPlot.
MINUNT(MXKYWD)	I*4	/INPUT2/	Minimum number of characters to accurately specify one of the MXKYWD units available on the Units page of CEdit. See INITUN for list of available units.
MNG	I*4	/PMWRT1/	**No longer used.**
MOUSIN	L*4	/MSKBD/	Logical indicating if mouse is installed.
MPP	I*4	/PMWRT1/	**No longer used.**
MPPP	I*4	/PMWRT1/	**No longer used.**
MPRATX	I*4	/MSKBD/	Distance in the X direction the mouse can move before the cursor is moved.
MPRATY	I*4	/MSKBD/	Distance in the Y direction the mouse can move before the cursor is moved.

Name	Type	Common Block	Description
MPRODR(NV,NS)	R*8	/MOCO1A/	Time history species production rates for a specified fire.
MPSDAT(3)	I*4	/MOCO1A/	Date of this run (see also CRDATE and RUNDAT)
MPSDATC	CHAR*8	/CONFG2/	Today's date in character format (see RUNDAT)
MSLB	L*4	/MSKBD/	True if left mouse button is pressed.
MSRB	L*4	/MSKBD/	True if right mouse button is pressed.
MSSIZE	I*4	/MSKBD/	Size in bytes of the mouse vector.
MVCALC	L*4	/MOCO1A/	Logical flag indicating where HVAC calculation is to be performed.
N	I*4	/MOCO1A/	Number of compartments in use (including the outside).
N2	I*4	/MOCO1A/	N+1.
N3	I*4	/MOCO1A/	2N+1. **No longer used.**
N4	I*4	/MOCO1A/	3N+1. **No longer used.**
NA(MBR)	I*4	/MOCO1A/	First node for HVAC branch IB (see NE).
NBR	I*4	/MOCO1A/	Number of HVAC branches.
NCNODE(MNODE)	I*4	/MOCO1A/	Number of nodes connected to HVAC node i.
NCNT(NRMX)	I*4	/MOCO2A/	Number of edges in each polygon.
NCOLS(4)	I*4	/MOCO2A/	Color value for each hazard level. See HZRDPL.
NCONFG	I*4	/MOCO1A/	1 if a graphics descriptor is present, 0 otherwise.
NDIV(MXSLB,NWAL,NR)	I*4	/MOCO1A/	Number of interior nodes in a wall material (of MXSLB slabs).
NDT	I*4	/MOCO1A/	Number of HVAC ducts.
NDUMPR	I*4	/MOCO1A/	1 if a history file specification is present, 0 otherwise.
NE(MBR)	I*4	/MOCO1A/	Last node for HVAC branch IB (see NA).
NEQUALS	I*4	/CENVIRO/	Total number of differential/algebraic equations that are solved. Note that only NODES equations are solved by DASSL. The rest are solved in SOLVE using Euler's method.

Name	Type	Common Block	Description
NEUTRAL(NR,NR)	I*4	/MOCO1A/	Number of neutral planes for a vent (Not very useful). **No longer used.**
NEXT	I*4	/MOCO1A/	Number of exterior nodes in the HVAC system.
NF(MBR)	I*4	/MOCO1A/	0 for duct, fan number for a fan - HVAC.
NFAN	I*4	/MOCO1A/	Number of fans in the HVAC system.
NFC(MFAN)	I*4	/MOCO1A/	Number of polynomial coefficients for HVAC fan i.
NFT	I*4	/MOCO1A/	Number of HVAC simple fittings.
NHVPVAR	I*4	/BLANK-COMMON/	Number of HVAC pressure unknowns. This number is the number of interior nodes or total nodes minus exterior nodes (NNODES-NEXT).
NHVSYS	I*4	/BLANK-COMMON/	Number of HVAC systems.
NHVTVAR	I*4	/BLANK-COMMON/	Number of HVAC temperature unknowns. This number is just the number of ducts (NBR).
NLIST(NTHMX)	CHAR*8	/MOCO1E/	Array of thermal property material names available in the current thermal database.
NLSPCT	I*4	/MOCO1A/	Number of species in this run.
NM1	I*4	/MOCO1A/	Actual number of compartments (N-1).
NMICKX	I*4	/MSKBD/	Number of GRAFIT ticks in the x direction.
NMICKY	I*4	/MSKBD/	Number of GRAFIT ticks in the y direction.
NNFILE	CHAR*60	/CONFG2/	Name of the input file (if appropriate).
NNODE	I*4	/MOCO1A/	Number of nodes in the HVAC system.
NOFHCL	I*4	/CENVIRO/	Solver array offset for HCl deposition variables.
NOFHVPR	I*4	/CENVIRO/	Solver array offset for HVAC species variables.
NOFP	I*4	/CENVIRO/	Solver array offset for compartment pressure variables.
NOFPMV	I*4	/CENVIRO/	Solver array offset for HVAC pressure variables.
NOFPRD	I*4	/CENVIRO/	Solver array offset for compartment species variables.

Name	Type	Common Block	Description
NOFSMK	I*4	/CENVIRO/	Solver array offset for smoke agglomeration variables.
NOFSMKW	I*4	/CENVIRO/	Solver array offset for soot deposition variables.
NOFTL	I*4	/CENVIRO/	Solver array offset for lower layer temperature variables.
NOFTMV	I*4	/CENVIRO/	Solver array offset for HVAC temperature variables.
NOFTU	I*4	/CENVIRO/	Solver array offset for upper layer temperature variables.
NOFVU	I*4	/CENVIRO/	Solver array offset for upper layer volume variables.
NOFWT	I*4	/CENVIRO/	Solver array offset for wall temperature variables.
NOPMX	I*4	/MOCO1A/	**No longer used.**
NP(NPLMX,NRMX)	I*4	/MOCO2A/	Pointer to the vertices of polynomial NRMX.
NPS(4,NRMX)	I*4	/MOCO2A/	Attributes (line width,fill,horizontal or vertical polynomial,polynomial number) of polynomial NRMX.
NRESTR	I*4	/MOCO1A/	Restart time (0 means no restart).
NRFLOW	I*4	/MOCO1A/	**No longer used.**
NSLB(NWAL,NR)	I*4	/MOCO1A/	Number of slabs for each specified surface in input file (see LNSLB).
NSMAX	I*4	/MOCO1A/	Maximum simulation time (seconds).
NTAB	I*4	/TRPTABL/	Table size for integration table used by the ceiling jet routine.
NUMNODE(MXSLB+1,4,NR)	I*4	/WNODES/	The number of nodes used to represent the wall temperature profiles for slab i in wall j in compartment k.
NUMOBJ	I*4	/OBJCT1/	Total number of objects in database + 1 (for default entry).
NUMOBJL	I*4	/OBJCT1/	Total number of valid objects from the input file.
NUMTHRM	I*4	/MOCO1D/	Total number of valid thermal property surfaces from the input file.

Name	Type	Common Block	Description
NVELEV	I*4	/VNTSLB/	Number of unique elevations (vent sill,soffit,layer height,neutral planes) where pressures are calculated to determine vent flow. Note the number of slabs is one less than NVELEV. **No longer used.**
NVENTS	I*4	/VENTS/	Total number of horizontal flow vents.
NVVENT	I*4	/VENTS/	Total number of vertical flow vents.
NW(NR,NR)	I*4	/MOCO1A/	Integer indicator for assigned numbers to horizontal flow vents between compartments (Coded for 1 to 4 by powers of 2).
NWALLS	I*4	/WNODES/	Total number of walls.
NWPTS	I*4	/WNODES/	Number of nodes used to represent the temperature in a wall (includes each wall slab).
NWV(NR,NR)	I*4	/MOCO1A/	Switch for vertical flow vent in a compartment (1 if present, 0 otherwise).
O2N2(NS)	R*8	/BLANK-COMMON/	Ratio of oxygen to nitrogen in the ambient.
OAREA(NV,MXOIN)	R*8	/OBJCT3/	Time history for area of object fire (m^2).
OBFILI	I*4	/CONFG2/	File unit number for accessing objects database.
OBJCNFG	CHAR*60	/CONFG2/	Objects database name from configuration file
OBJCR1(3,MXOIN)	R*8	/MOCO1A/	Element 1-Start time, **others no longer used. Should be replaced by OBJCRI.**
OBJCRI(3,MXOIN)	R*8	/OBJCT3/	Element 1-Start time, 2-Flux for ignition (w/m^2), 3-Surface temperature for ignition (k).
OBJDEF(MXOIN)	L*4	/OBJCT3/	Logical indicating if object entry definition in database is complete (see OBJLD). **May be obsolete.**
OBJGMW(MXOIN)	R*8	/OBJCT3/	Object gram molecular weight from database.
OBJHC(NV,MXOIN)	R*8	/OBJCT3/	Time history for object heat of combustion (J/kg) from database.
OBJIGN(MXOIN)	I*4	/MOCO1A/	First element on object to begin burning from input file.
OBJLD(MXOIN)	L*4	/OBJCT1/	Logical indicating if object specified in input file is valid (see OBJDEF). **May be obsolete.**

Name	Type	Common Block	Description
OBJLFM(MXOIN)	I*4	/OBJCT3/	Number of points in an object time history specification in database.
OBJMA1(MXOIN)	R*8	/MOCO1A/	**No longer used.**
OBJMAS(MXOIN)	R*8	/OBJCT3/	Object total mass (kg) from database.
OBJNAM(MXOIN)	CHAR*60	/OBJCT2/	Object name from the object database for a specified object in CFAST input file.
OBJNIN(MXOIN)	CHAR*60	/OBJCT0/	Object name as specified in the input file.
OBJON(MXOIN)	L*4	/MOCO1A/	Logical indicating for each specified object if conditions have been met for the object to begin burning.
OBJORT(3,MXOIN)	R*8	/OBJCT3/	Object orientation angles PHI, THETA, PSI from database.
OBJPNT(MXOIN)	I*4	/OBJCT1/	Contains the element number within the ODBREC array for each valid object specified in the input file. GETOBJ uses this element number to determine the record number to read from the database.
OBJPOS(3,MXOIN)	R*8	/MOCO1A/	Position of an object within the specified compartment: element 1-depth(X), 2-breadth(Y), 3-height(Z) from input file.
OBJRM(MXOIN)	I*4	/MOCO1A/	Object compartment from input file.
OBJTYP(MXOIN)	I*4	/OBJCT3/	Object type from database.
OBJVT(MXOIN)	R*8	/OBJCT3/	Object volatilization temperature (k) from database.
OBJXYZ(4,MXOIN)	R*8	/OBJCT3/	Object panel length (m), panel height or width (m), panel thickness (m), surface area of panel element (m^2) from database.
OCO(NV,MXOIN)	R*8	/OBJCT3/	Time history for object CO/CO_2 from database.
OCRATI(NV)	R*8	/MOCO1A/	Time history of fuel hydrogen/carbon ratio.
ODBNAM(MXOBJ)	CHAR*60	/OBJCT0/	Array of all object names available in the current object database (see ODBREC).
ODBREC(MXOBJ)	I*4	/OBJCT1/	Starting record number by object for accessing detailed information in the object database (see ODBNAM).

Name	Type	Common Block	Description
OFFSTD(MXUNT)	R*4	/INPUT2/	Offsets used in converting from user specified units to scientific units for each of the MXUNT derived quantities such as volume, density, specific heat, etc. See CUNITS for list of derived quantities.
OFFSTU(MXUNT)	R*4	/INPUT2/	Offsets used in converting from scientific units to user specified units for each of the MXUNT derived quantities such as volume, density, specific heat, etc. See CUNITS for list of derived quantities. Used in CEdit to display.
OFILE	CHAR*60	/CONFG2/	Name of the objects database to be used in verifying selected objects. Obtained from input file or configuration file.
OHCR(NV,MXOIN)	R*8	/OBJCT3/	Object H/C time history from database.
OHIGH(NV,MXOIN)	R*8	/OBJCT3/	Time history for object height (m) from database.
OIFIRED(MXOIN)	I*4	/MOCO1A/	**No longer used.**
OLDMS(MSBYTE/4)	I*4	/MSKBD/	Temporarily store mouse vector array.
OMASS(NV,MXOIN)	R*8	/OBJCT3/	Time history of object pyrolysis rates (kg/sec) from database.
OMPRODR(NV,NS,MXOIN)	R*8	/OBJCT3/	Species production rate history for object. Element 10-CT time history, 5-HCN time history, 6-HCL time history from database.
ONTARGET(NR)	R*8	/MOCO1A/	Absolute radiation from the upper layer to a target (less ambient).
OOC(NV,MXOIN)	R*8	/OBJCT3/	Object O/C time history from database.
OOD(NV,MXOIN)	R*8	/OBJCT3/	Object OD or soot time history from database.
OPENHLP	L*4	/INPUT2/	Logical indicating if help file is open for CEdit.
OPLUME(3,MXOIN)	R*8	/MOCO1A/	Pyrolysis rate, plume entrainment rate, and plume flow rate for the “other objects.”
OPNOBJ	L*4	/INPUT2/	Logical indicating if the objects database has been opened.
OPTION(MXOPT)	I*4	/OPT/	Logical flags to turn on or off individual physical phenomena (1-on,0-off).
OPTIONS(MOPT)	CHAR*4	/CONFG2/	Command line options up to a maximum of MOPT.

Name	Type	Common Block	Description
OQDOT(NV,MXOIN)	R*8	/OBJCT3/	Time history for object rate of heat release (w) from database.
OTFIRET(MXOIN)	R*8	/MOCO1A/	**No longer used.**
OTFMAXT(MXOIN)	R*8	/OBJCT3/	Value of the last time point in an object time history specification in database.
OTIME(NV,MXOIN)	R*8	/OBJCT3/	Object time history points (sec) from database.
OUTFILE	CHAR*60	/CONFG2/	Name of the output file (if appropriate).
OUTPUT(4096)	I*4	/MOCO1A/	Equivalenced with GAMMA in DUMPER.
P(NT)	R*8	/MOCO1A/	Solution vector of pressure, upper and lower temperature, volume.
PA	R*8	/MOCO1A/	Internal ambient pressure at the reference elevation.
PACOLOR(4,5)	I*4	/MOCO2A/	Color numbers for the 5 graphics palettes.
PALABEL(4,5)	CHAR*50	/MOCO2B/	Titles for the 5 graphics palettes.
PALETT(6,5)	R*4	/MOCO2A/	X,Y,Z positions for the 5 graphics palettes.
PAMB(NR)	R*8	/MOCO1A/	Ambient pressure in a compartment prior to the fire.
PASSFILE	CHAR*60	/CONFG2/	Used by the shell routine to pass the environment to CFAST, CPlot, ...
PATH	CHAR*64	/CONFG2/	Path searched for the data files.
PATHSOL	CHAR*948	/CONFG2/	Array of character values read from PASSFILE. Equivalenced to PATH in READOP.
PCO2LO	R*8	/PMWRT1/	**No longer used.**
PCO2ST	R*8	/PHYCNS/	Stoichiometric pressure of CO_2 in flame (FSM - not implemented).
PCO2UP	R*8	/PMWRT1/	**No longer used.**
PDEL	R*8	/PHYCNS/	Characteristic flame length (FSM - not implemented).
PDOLD(MAXTEQ)	R*8	/DERVS/	Derivative of solution at “old” or last successfully completed time step.
PFACTS(MXPRD,MXFIRE)	R*8	/PRODS/	**No longer used.**
PFILE	CHAR*60	/CONFG2/	File name for partition database.

Name	Type	Common Block	Description
PH2OLO	R*8	/PMWRT1/	**No longer used.**
PH2OST	R*8	/PHYCNS/	Stoichiometric pressure of H_2O in flame (FSM - not implemented).
PH2OUP	R*8	/PMWRT1/	**No longer used.**
PINIT(NT)	R*8	/SOLVPRM/	Transient solver array, will be obsolete.
PLEN	I*4	/CONFG1/	Number of characters in the data path - zero implies not used.
PLUME(6,5)	R*4	/MOCO2A/	**No longer used.**
PNLDS	R*8	/OBJCT3/	Distance between sides of an object panel (m) from database.
POFSET	R*8	/MOCO1A/	Pressure offset to help solve the stiffness problem.
POINT(190)	I*4	/INPUT2/	Record pointers to entries in the help file.
POLD(MAXTEQ)	R*8	/DERVS/	Solution at "old" or last successfully completed time step. POLD and PDOLD are used only with the step size DT to update solution variables that are not solved for by DASSL.
POSTN(NVAR)	I*4	/PLTPAR/	**No longer used.**
PPMDV(2,NR,NS)	R*8	/MOCO1A/	Mass concentration (kg/m^3).
PRDS0(MXIRM,MXPRD,2)	R*8	/PRODS/	**No longer used.**
PREF	R*8	/MOCO1A/	Default reference pressure (1.03 e+5).
PREVNFIL	CHAR*60	/CONFG2/	Name of input file when first opened in CEdit. Allows save to a file with another name or an exit without saving and updating the configuration file.
PRMOLD(NS)	R*8	/COMINT/	**No longer used.**
PRODMS(NS)	R*8	/COMINT/	**No longer used.**
PRODUP	L*4	/DERVS/	**No longer used.**
QC(2,NR)	R*8	/MOCO1A/	Net convective heat loss from a zone (Watts).
QCFOLD	R*8	/COMINT/	**No longer used.**
QCFOOT	R*8	/PHYCNS/	Convective heat flux at flame front (FSM - not implemented).
QCFURN	R*8	/COMINT/	**No longer used.**

Name	Type	Common Block	Description
QCVENT(NR,NR,4,NV)	R*8	/BLANK-COMMON/	Opening parameter for a vent expressed as a fraction between 0.0 and 1.0.
QDIN(NWAL,NR)	R*8	/BLANK-COMMON/	Net heat gain to an interior side (exposed) of a wall (w/m^2). **No longer used.**
QDOUT(NWAL,NR)	R*8	/BLANK-COMMON/	Net heat loss on the back side of a wall (w/m^2). **No longer used.**
QF(NR)	R*8	/MOCO1A/	Net heat generation rate of a fire into a zone (Watts).
QFC(2,NR)	R*8	/BLANK-COMMON/	Total convective heat gain (+) or loss (-) by a layer.
QFIRED(NV)	R*8	/MOCO1A/	Time history of heat release rate for a specified fire.
QFR(MXFIRE)	R*8	/BLANK-COMMON/	Total radiative heat gain (+) or loss (-) by a layer.
QGAS(3)	R*8	/COMINT/	**No longer used.**
QGSOLD(3)	R*8	/COMINT/	**No longer used.**
QMAX(MFAN)	R*8	/MOCO1A/	Flow rate at HMAX(k).
QMIN(MFAN)	R*8	/MOCO1A/	Flow rate at HMIN(k).
QR(2,NR)	R*8	/MOCO1A/	Net radiative loss from a zone (Watts).
QRADRL	R*8	/MOCO1A/	Fraction of heat which leaves a fire as radiation.
QRWALL(4)	R*8	/COMINT/	**No longer used.**
QRWOLD(4)	R*8	/COMINT/	**No longer used.**
QSCNV(NWAL,NR)	R*8	/BLANK-COMMON/	Convective flux to a wall (w/m^2).
QSRADW(NWAL,NR)	R*8	/BLANK-COMMON/	Radiative flux to a wall (w/m^2). **No longer used.**
QZERO	R*8	/PHYCNS/	Scaling heat and mass parameter (FSM - not implemented).
RA	R*8	/MOCO1A/	Default internal ambient density (kg/m^3) at reference elevation.
RAMB(NR)	R*8	/MOCO1A/	Initial ambient mass density in a compartment
RCNT(5)	I*4	/MOCO2A/	Number of variables in 5 DISPLAYC tables.
RELHUM	R*8	/MOCO1A/	Initial relative humidity (default 0 percent).

Name	Type	Common Block	Description
REPORTO	I*4	/CONFG1/	REPORTG - time delay; REPORT - type of output
RFILE	CHAR*60	/CONFG2/	Restart file.
RGAS	R*8	/MOCO1A/	Universal gas constant.
RHVPTOL	R*8	/SOLVPRM/	Relative error tolerance for pressure at interior HVAC nodes.
RHVTTOL	R*8	/SOLVPRM/	Relative error tolerance for temperature in HVAC ducts.
RLHEAT	R*8	/COMINT/	**No longer used.**
RLHOLD	R*8	/COMINT/	**No longer used.**
RLMASS	R*8	/COMINT/	**No longer used.**
RLMOLD	R*8	/COMINT/	**No longer used.**
RMAX	R*8	/TRPTABL/	Maximum radial distance from center of plume to compartment boundary.
RMFILE	I*4	/CONFG1/	Indicates local or remote path (obsolete).
RMINDX	I*4	/INPUT2/	Compartment number displayed as the leftmost compartment on the geometry screen in CEdit. Used on vents, thermal properties, and objects screens to determine which compartments display.
RMTMP	R*8	/PHYCNS/	FSM - not implemented.
RMTMPB	R*8	/PHYCNS/	FSM - not implemented.
ROHB(MBR)	R*8	/MOCO1A/	Density of gas in HVAC branch IB.
ROOM(NRMX)	I*4	/MOCO2A/	Compartment number to which polygon NRMX is mapped.
RPAR2(1)	R*8	/SOLVPRM/	Used to pass pressure error tolerances to RESID.
RPTOL	R*8	/SOLVPRM/	Relative error tolerance for pressures.
RR(MDT)	R*8	/MOCO1A/	Relative roughness of walls of HVAC duct id.
RSTOCH	R*8	/PHYCNS/	Stoichiometric fuel/air ratio (FSM - not implemented).
RTOL	R*8	/SOLVPRM/	Relative error tolerances for layer volumes and temperatures.
RTYPE(NRMX)	I*4	/MOCO2A/	The type of polygon NRMX. Type 2 is a vertical polygon used to represent the upper layer.

Name	Type	Common Block	Description
RUNDAT(3)	I*4	/CONFG1/	Today's date (day, month, year). RUNDAT is copied to MPSDAT in INITFS or CFAST as soon as the model kernel is started.
RW(MXSLB,NWAL,NR)	R*8	/MOCO1A/	Material density of a boundary slab (kg/m^3) for each specified surface in input file (see LRW).
RWTOL	R*8	/SOLVPRM/	Relative error tolerances for wall temperatures.
SA(NR,NR,4)	R*8	/MOCO1A/	Flow field upper to lower (kg/s).
SAL	R*8	/MOCO1A/	Reference elevation (m) - default to zero.
SAU(NR,NR,4)	R*8	/MOCO1A/	Entrainment rate into the upper layer.
SAVECL(11)	I*4	/INPUT2/	Save initial color settings so that CEdit can implement the reset option on the Version and Settings screen
SAVEINPT	L*4	/INPUT2/	Logical indicating if changes have been made in current session with CEdit.
SAVETHRM	L*4	/INPUT2/	**No longer used.**
SAVEUN(7)	I*4	/INPUT2/	Save initial unit settings so that CEdit can implement the reset option on the Version and Settings screen.
SCAT	R*8	/PHYCNS/	Scattering coefficient of smoke (FSM - not implemented).
SGFIRE	R*8	/COMINT/	**No longer used.**
SHELL	L*4	/CONFG1/	Logical, set to true if the shell is doing the procedure.
SIGAVG	R*8	/COMINT/	**No longer used.**
SIGE	R*8	/PHYCNS/	Specific soot extinction area (FSM - not implemented).
SIGM	R*8	/MOCO1A/	Stefan-Boltzmann constant ($5.67 \cdot 10^{-8}$ $W/m^2/K^4$)
SIGMAG	R*8	/PHYCNS/	Pool/wall flame critical surface gradient (FSM - not implemented).
SKMAX	R*8	/PHYCNS/	Maximum soot absorption coefficient (FSM - not implemented).
SMKAGL	I*4	/MOCO1A/	Switch for smoke agglomeration (0-off,1-on).
SS(NR,NR,4)	R*8	/MOCO1A/	Flow field from upper to upper layer (kg/s).

Name	Type	Common Block	Description
SSAVG	R*8	/PMWRT1/	**No longer used.**
SSPEAK	R*8	/PMWRT1/	**No longer used.**
SSQPEK	R*8	/PMWRT1/	**No longer used.**
START	I*4	/READ1C/	Position within INBUF string to begin parsing for a space delimited substring; used by READIN, etc. (see FIRST, LAST, START)
STIME	R*8	/MOCO1A/	Current simulation time (s) - corresponds to ITMSTP.
STPMAX	R*8	/SOLVPRM/	Maximum solver step size.
STYPE(NS)	CHAR*4	/DFLTS/	Name for each of the NS available species.
SWITCH(NWAL,NR)	L*4	/MOCO1A/	Logical switch for wall conduction (...,NR) used for output.
TA	R*8	/MOCO1A/	Reference ambient temperature (K).
TABL(100)	R*8	/TRPTABL/	Table of integral values for heat transfer parameters used in ceiling jet algorithm.
TABLE(6,5)	R*4	/MOCO2A/	X,Y,Z positions for the 5 graphics tables.
TAMB(NR)	R*8	/MOCO1A/	Ambient temperature in a compartment (K).
TBR(MBR)	R*8	/MOCO1A/	Absolute temperature of gases in HVAC branch IB.
TBURN	R*8	/PHYCNS/	Surface temperature of burning surface (FSM - not implemented).
TC	R*8	/AINTCH/	Intermediate value for ceiling jet routine.
TE	R*8	/MOCO1A/	Pyrolysis temperature of the fuel.
TELLEN(126)	I*4	/INPUT2/	Array of string lengths for each element in TELTIT array.
TELTIT(126)	CHAR*20	/INPUT1/	Array of display strings for screens in CEdit.
TEPM	R*8	/COMINT/	**No longer used.**
TERMXX	R*8	/MOCO1A/	Dummy variable to indicate storage location of the end of the CFAST COMMON block. Used to calculate the size of the common block in bytes.
TERRORS(NTHMX)	CHAR*8	/MOCO1E/	Code for errors in the thermal database.
TFIRED(NV)	R*8	/MOCO1A/	Time point specifications.

Name	Type	Common Block	Description
TFIRET	R*8	/MOCO1A/	Current time for interpolation. **No longer used.**
TFMAXT	R*8	/MOCO1A/	Value of the last time point for the specified fire.
TGIGNT	R*8	/MOCO1A/	Ignition temperature for a well stirred gas - limits fires in vents.
THDEF(NTHMX)	L*4	/MOCO1D/	Logical indicating if thermal property material name from database is correctly defined (see THSET).
THRMCNFG	CHAR*60	/CONFG2/	Name of thermal properties database from configuration file.
THRMFILE	CHAR*60	/CONFG2/	Name of the thermal database to be used in verifying materials. Obtained from input file or configuration file.
THRMIA	R*8	/PHYCNS/	Thermal inertia (FSM - not implemented).
THRMLT	R*8	/PHYCNS/	Thermal thickness (FSM - not implemented).
THSET(NWAL,NR)	L*4	/MOCO1D/	Logical indicating if specified material is correctly defined in the thermophysical properties database (see THDEF).
THT	R*8	/AINTCH/	Temperature of ambient in unconfined ceiling heat transfer problem for ceiling jet calculation (upper/lower layer temperature).
THTQHP	R*8	/AINTCH/	Enthalpy of ambient in unconfined ceiling heat transfer problem for ceiling jet calculation (upper/lower layer temperature).
TIG	R*8	/PHYCNS/	Piloted ignition temperature (FSM - not implemented).
TIK(3)	I*4	/MOCO2A/	Decoded simulation time label (HH|MM|.SS) in DISPLAYC.
TITLE(50)	CHAR*1	/MOCO1B/	Title from the VERSN line in input file.
TITLES(NVAR)	CHAR*48	/PLTPAR/	Description of physical variables accessible in CPlot.
TOLD	R*8	/DERVS/	Time at last time step.
TOXICT(2,NR,NS)	R*8	/MOCO1A/	Conglomeration of stuff for output.
TREF	R*8	/MOCO1A/	Default reference temperature.
TRNFRM(16,5)	R*4	/MOCO2A/	Transformation matrix for a view in a graphics display.

Name	Type	Common Block	Description
TWE(NWAL,NR)	R*8	/MOCO1A/	Temperature of the gas external to a compartment boundary.
TWJ(NN,NR,NWAL)	R*8	/MOCO1A/	Temperature profile in the boundaries (ceiling, floor, upper/lower wall).
TYPE	I*4	/READ1C/	Type of variable entered from keyboard. Determined by CONVRT (1-integer, 2-real, 3-character).
TYPEUN(7)	I*4	/INPUT2/	User selected units for each of the base quantities: temperature, pressure, length, energy, mass, time, and time.
UFACTR(MXKYWD)	R*4	/INPUT2/	Conversion factors to convert each of the MXKYWD available units to scientific units for the associated base quantity. See INITUN for list of available units.
UNIT(NVAR)	I*4	/PLTPAR/	Unit type for each of the NVAR variables in CPlot.
UNITS(7)	I*4	/CONFG1/	User selected units for each of the base quantities: temperature, pressure, length, energy, mass, time, and time. READCF reads from the configuration file or sets to default values and then sets TYPEUN.
UNTKWD(MXKYWD)	CHAR*10	/INPUT1/	Keyword for each of the MXKYWD available units. Used with MAXUNT and MINUNT to uniquely identify a user specified unit. Used for display in CEdit and CF_Set and on the Units page of CEdit. See INITUN for list of available units.
UNTMAP(MXKYWD)	I*4	/INPUT2/	Maps each of the MXKYWD available units to the appropriate base quantity: temperature, pressure, length, energy, mass, time, and time. See INITUN for list of available units and mapping assignments. See SETUNT for numeric representation of the base quantities.
UOFSTR(MXKYWD)	R*4	/INPUT2/	Offsets used to convert each of the MXKYWD available units to scientific units for the associated base quantity. See INITUN for list of available units.
UORL(16,25)	CHAR*1	/MOCO2B/	CPlot - upper or lower layer.
USDFSM	L*4	/OBJCT1/	Logical indicating that an object of type FSM has already been selected. Only one is allowed.
VAA(2,MXVENT)	R*8	/VENTS/	Mass flow from lower layer to lower layer in a natural flow vent.

Name	Type	Common Block	Description
VALID	L*4	/READ1C/	Logical indicating if substring parsed from INBUF string is valid (see FIRST, LAST, etc.)
VAS(2,MXVENT)	R*8	/VENTS/	Mass flow from lower layer to upper layer in a natural flow vent.
VASA(2,MXVENT)	R*8	/VENTS/	Entrained mass flow from upper layer to lower layer in a natural flow vent.
VERSION	I*4	/CONFG1/	Actual working version of the model. Copied to IVERS by INITFS or CFAST.
VIEU(6,5)	R*4	/MOCO2A/	The dimensions in 3D space for a picture in the graphics window.
VIEWFLE(5)	CHAR*17	/MOCO2B/	Picture files used by CFAST.
VOLFRL(NR)	R*8	/BLANK-COMMON/	V_l/V - lower volume fraction.
VOLFRU(NR)	R*8	/BLANK-COMMON/	V_u/V - upper volume fraction.
VR(NR)	R*8	/MOCO1A/	Volume of a compartment.
VSA(2,MXVENT)	R*8	/VENTS/	Mass flow from upper layer to lower layer in a natural flow vent.
VSAS(2,MXVENT)	R*8	/VENTS/	Entrained mass flow from lower layer to upper layer in a natural flow vent.
VSHAPE(NR,NR)	I*4	/MOCO1A/	Shape of a vertical flow vent (1 - Circle, 2 - Square).
VSS(2,MXVENT)	R*8	/VENTS/	Mass flow from upper layer to upper layer in a natural flow vent.
VVAREA(NR,NR)	R*8	/MOCO1A/	Area of a vertical flow vent.
WALLDX(NN,NR,4)	R*8	/WNODES/	WALLDX(i,j,k) is the distance between nodes i and i+1 in wall k of compartment j.
WINDC(NR)	R*8	/MOCO1A/	Wind coefficient for a vent facing the outside.
WINDOW(6)	R*4	/MOCO2A/	The dimensions for the graphics window used by CFAST.
WINDPW	R*8	/MOCO1A/	Wind power law coefficient.
WINDRF	R*8	/MOCO1A/	Wind reference height (m).
WINDV	R*8	/MOCO1A/	Wind reference velocity at WINDRF.
WLENGTH(NR,4)	R*8	/WNODES/	WLENGTH(i,j) is the thickness (m) of wall j in compartment i.
WSPLIT(3)	R*8	/WNODES/	The fraction of wall nodes to occur in slab i.

Name	Type	Common Block	Description
XF	R*8	/AINTCH/	X coordinate of the base of the fire in the ceiling jet algorithm.
XI	R*4	/READ1C/	Real numeric entry from keyboard.
XLAM	R*8	/PHYCNS/	Laser wavelength of soot optical probe in cone calorimeter data for object fires (FSM - not implemented).
XPALET(4)	R*4	/MOCO2A/	The minimum height above the floor for each hazard level (see HZRDPL).
XYCOOD(4)	R*4	/DEVTYP/	Conversion factors for coordinates in the graphics display. Pulled in by GRAFIT.SOR.
XYZ(3,NPP)	R*4	/MOCO2A/	X,Y,Z coordinates of the points used by CFAST graphics.
YF	R*8	/AINTCH/	Y coordinate of the base of the fire in the ceiling jet algorithm.
YINTER(NR)	R*8	/BLANK-COMMON/	Equivalenced with QFR in NPUTP.
YVELEV(10)	R*8	/VNTSLB/	Elevations of the ends of the flow slabs in the horizontal flow calculations.
ZZCSPEC(NR,2,NS)	R*8	/CENVIRO/	Mass fraction of a species in a layer of a compartment.
ZZGMAX(NR)	R*8	/CENVIRO/	Maximum gas species amount in compartment i.
ZZGMIN(NR)	R*8	/CENVIRO/	Minimum gas species amount in compartment i.
ZZGSPEC(NR,2,NS)	R*8	/CENVIRO/	ZZGSPEC(i,j,k) is the amount of species k in layer j in compartment i.
ZZHLAY(NR,2)	R*8	/CENVIRO/	ZZHLAY(i,j) is the height of layer j in compartment i.
ZZHVM(MXHVSYS)	R*8	/CENVIRO/	Mass of gas in HVAC system ISYS.
ZZHVPR(MXHVSYS,NS)	R*8	/CENVIRO/	Amount of product IPROD associated with HVAC system ISYS.
ZZMASS(NR,2)	R*8	/CENVIRO/	ZZMASS(i,j) is the mass of layer j in compartment i.
ZZPABS(NR)	R*8	/CENVIRO/	Absolute pressure in compartment i.
ZZPMAX(NR)	R*8	/CENVIRO/	Maximum pressure in compartment i. **No longer used.**
ZZPMIN(NR)	R*8	/CENVIRO/	Minimum pressure in compartment i. **No longer used.**
ZZQ(NR,2)	R*8	/CENVIRO/	ZZQ(i,j) is the energy of layer j in compartment i.

Name	Type	Common Block	Description
ZZRELP(NR)	R*8	/CENVIRO/	Pressure relative to POFSET in compartment i.
ZZRHO(NR,2)	R*8	/CENVIRO/	ZZRHO(i,j) is the density of layer j in compartment i.
ZZTEMP(NR,2)	R*8	/CENVIRO/	ZZTEMP(i,j) is the temperature of layer j in compartment i.
ZZTMAX(NR)	R*8	/CENVIRO/	Maximum temperature in compartment i. **No longer used.**
ZZTMIN(NR)	R*8	/CENVIRO/	Minimum temperature in compartment i. **No longer used.**
ZZVENT(MXVENT,3)	R*8	/VENTS/	ZZVENT(i,j) where j=1-height of sill(bottom) relative to floor for vent i,2-height of soffit(top) relative to floor for vent i,3-width of vent i.
ZZVMAX(NR)	R*8	/CENVIRO/	Maximum volume in compartment i.
ZZVMIN(NR)	R*8	/CENVIRO/	Minimum volume in compartment i.
ZZVOL(NR,2)	R*8	/CENVIRO/	ZZVOL(i,j) is the volume of layer j in compartment i.
ZZWMAX(NR)	R*8	/CENVIRO/	Maximum wall species amount in compartment i.
ZZWMIN(NR)	R*8	/CENVIRO/	Minimum wall species amount in compartment i.
ZZWSPEC(NR,NWAL)	R*8	/CENVIRO/	ZZWSPEC(i,j) is the concentration of species (currently HCl) in wall j in compartment i.
ZZWTEMP(NR,NWAL,2)	R*8	/CENVIRO/	Wall temperature of a wall surface in a compartment. 1-inside temperature,2-outside temperature.
ZZYCEIL(NR)	R*8	/CENVIRO/	Height of the ceiling in compartment i. **No longer used.**
ZZYFLOR(NR)	R*8	/CENVIRO/	Height of the floor above the reference elevation in compartment i.

Appendix D: CFAST Common Blocks

Name	Size	Declaration	Members
AINTCH	80	CEILHT.SOR in routines CEILHT, QFCLG	H HTCT THT THTQHP C1 C2 C3 XF YF TC
BLANK-COMMON	186820	PARAMS.INS	EXSAL QFR QFC QSCNV QDOUT QSRADW HMFLOW MAPLTW QDIN EXPA EXTA EXRA QCVENT O2N2 HWJDOT HTOT HTFLOW HTFNET VOLFRU VOLFRL HVFRAC HCRATT IHMLAR HVMFSYS DHVPRSYS HVTM HVDARA HVT CHV DUCTCV EXSET ALLOWED IZHVMAPI IZHVMAPE IZHVIE NHVPVAR NHVTVAR IZHVSYS IZHVBSYS NHVSYS
CENVIRO	10964	CENVIRO.INS	ZZTMIN ZZTMAX ZZVMIN ZZVMAX ZZPMIN ZZPMAX ZZGMAX ZZGMIN ZZWMAX ZZWMIN ZZVOL ZZHLAY ZZRELP ZZPABS ZZTEMP ZZRHO ZZMASS ZZQ ZZGSPEC ZZCSPEC ZZWSPEC ZZYFLOR ZZYCEIL ZZWTEMP ZZHVPR ZZHVM IZPMAP NOFP NOFPMV NOFTMV NOFTU NOFVU NOFTL NOFWT NOFPRD NOFHCL NOFSMKW NOFSMK NOFHVPR NEQUALS
COMINT	384	BLOCKS.INS	ETIME DELTE QRWOLD QRWALL QGSOLD QGAS RLHOLD RLHEAT QCFOLD QCFURN HFTOLD HFOTOP RLMOLD RLMASS SGFIRE SIGAVG PRMOLD PRODMS TEPM APSPM
CONFG1	136	CSHELL.INS	ICHDR ICSUB ICTXT ICPRO ICMSG ICHLP ICBG ICMBG ICHBG ICEBG ICEMS UNITS VERSION ADVFEA SHELL DLEN PLEN RUNDAT LOGERR CSOPEN RMFILE CF_EXIST HEADER IOFILI IOFILO REPORTO
CONFG2	1132	CSHELL.INS	PATH DPATH CURRENT CONFIG THRMFILE DFILE GFILE OFILE PFILE OUTFILE NNFILE CARDTYPE OPTIONS RFILE MPSDATC DUMPF PASSFILE ADUMPF PREVNFIL THRMCNFG OBFILI OBJCNFG
DEBUG	80	OPT.INS	DEBUG
DERVS	12804	DERVS.INS	POLD PDOLD TOLD DT PRODUP
DEVTYP	32	GRAFIT.SOR	IDEVIC LSW LTSW XYCOOD LUOUT
DFLTS	68	DISPLAYC.SOR INIPAR.SOR INITDI.SOR NPUTG.SOR	IDEF ITODEF LAYDEF ISPDEF ICHRS IDEVO IDEVC STYPE
GRFTYP	56	GRAFIT.SOR WDDRAW.SOR	ANGLE IRVRSE CHSIZE ITKWIT LUDIAG LUHSET
INPUT1	3223	CFIN.INS	TELTIT LBUF CBUF UNTKWD LOGTIM
INPUT2	2144	CFIN.INS	TELLEN MAXTIT LCROW LCCOL LBACK LFORE LRON POINT OPENHLP GTRAIL FASTFOR SAVEINPT SAVETHRM SAVECL SAVEUN FACTRU OFFSTU FACTRD OFFSTD TYPEUN UNTMAP UFACTR UOFSTR MINUNT MAXUNT RMINDX OPNOBJ

Name	Size	Declaration	Members
MOCO1A	159896	CFAST.INS	GAMMA G SIGM CP TA RA PREF RGAS POFSET PA TREF SAL SS SA AS AA SAU ASL MINMAS QF QR QC HEATUP HEATLP HEATVF EMP EMS EME APS VVAREA QRADRL HCOMBA HVDELT HVGRAV HVRGAS CE HWJ BFIRED AFIRED HFIRED TFIRED TFMAXT TFIRET MPRODR MFIRET P BW HH HL WINDC CNVG HHP HLP WINDV WINDRF WINDPW DELTAT TGIGNT STIME LIMO2 RELHUM GMWF HOCBMB COCO2 HVEFLO BR DR HR AR HRP VR HRL NSLB TE PPMDV TAMB RAMB PAMB ETA ERA FKW CW RW EPA FLW EPW QFIRED TWJ TWE HFLR ONTARGET CCO2 MASS TOXICT HCRATIO HVEXTT AREXT HVORIEN HVELXT HVDVOL TBR ROHB BFLO IBRF CO AO HVP HVGHT DPZ HVFLOW QMAX HMIN HMAX HVBCO DFMIN DFMAX QMIN DE DA DL RR DUCTAR HVCONC HVEXCN HCLBF OCRATI OBJCR1 OBJPOS IBRD OTFIRET OBJMA1 OPLUME HMFNET FPOS HCNF HCLF TERMXX NEUTRAL NLSPCT IVERS LFMAX LFBO LFBT SWITCH IFIRED NOPMX NRFLOW LPRINT NSMAX LDIAGP LDIAGO ITMMAX MAXINR IDIAG NM1 N N2 N3 N4 ITMSTP NDIV ACTIVS NCONFG NDUMPR NRESTR LCOPY CRDATE MPSDAT NDT NEXT NA NE NF MVCALC NNODE NFT NFAN NBR NCNODE IN NFC NWV NW HVNODE ICMV SMKAGL OBJIGN OBJON CJETON HCLDEP OBJRM VSHAPE OIFIRED ITERMXX
MOCO1B	50	CFAST.INS	TITLE
MOCO1D	9672	THERMP.INS	LFKW LCW LRW LFLW LEPW LHCLBF THDEF MAXCT LNSLB THSET NUMTHRM
MOCO1E	1408	THERMP.INS	CNAME NLIST TERRORS
MOCO2A	37564	DISPLAY.INS	PACOLOR GASPTR TABLE PLUME BAR PALETT GRCOLR GRAPH LABCOLR LABLE VIEU LABANGL TRNFRM MAXIND WINDOW CNUM GASN RCNT NCOLS XYZ RTYPE NCNT ROOM TIK EDG EDGS NP NPS HZRD HZRDPL INDEX GASCNT XPALET LEGEND LESPEC LOGOON LEGENDON AHISTOV AHISTI
MOCO2B	4935	DISPLAY.INS	PALABEL LABTEXT VIEWFLE UORL BART GRFT
MSKBD	772	CFIN.INS	KBPNT KBCNT KBXBF KBINS KBSET KBLNGT KBSTRT KBFRST KBLAST MOUSIN NMICKY NMICKX MPRATX MPRATY MSLB MSRB MDELAY OLDMS MSSIZE
OBJCT0	2760	OBJECTS1.INS	ODBNAM OBJNIN
OBJCT1	260	OBJECTS1.INS	OBJPNT ODBREC NUMOBJ NUMOBJL OBJLD USDFSM
OBJCT2	1260	OBJECTS2.INS	OBJNAM CONEF

Name	Size	Declaration	Members
OBJCT3	55756	OBJECTS2.INS	OTIME OBJCRI OBJMAS OBJGMW OBJHC OBJVT OMASS OAREA OHIGH OQDOT OCO OHCR OOD OOC OMPRODR OTFMAXT OBJXYZ OBJORT PNLDS OBJLFM OBJTYP OBJDEF CONEN
OPT	80	OPT.INS	OPTION
PHYCNS	232	BLOCKS.INS	DTIME CTIME EMISIV AIRAK RSTOCH RMTMP RMTMPB THRMLT THRMIA PDEL TIG QCFOOT FOOTLT QZERO SKMAX XLAM SIGE ISTOCH IHIST CQ DQ DENAIR HCPAIR TBURN SCAT FLMTMP SIGMAG PCO2ST PH2OST IRMTRM IPLOT
PLTPAR	2604	DISPLAYC.SOR INIPAR.SOR OUTPU1.SOR	MAXSZ MINSZ POSTN COFSET LOFSET UNIT LABEL TITLES KEYWRD
PMWRT1	80	BLOCKS.INS	PCO2LO PCO2UP PH2OLO PH2OUP MNG MPP MPPP IWRITE SSPEAK SSQPEK SSAVG FSAVG
PRODS	4880	PRODS.INS	PFACTS PRDS0 ACON
READ1C	32	CFIO.INS	COUNT START FIRST LAST IX TYPE XI VALID
READ2C	256	CFIO.INS	INBUF
SOLVPRM	4216	SOLVPRM.INS	APTOL RPTOL ATOL RTOL AHVPTOL RHVPTOL AHVTTOL RHVTTOL AWTOL RWTOL ALGTOL STPMAX DASSLFTS PINIT RPAR2 IPAR2
TRPTABL	812	CEILHT.SOR in routines MAKTABL, INTTABL	TABL RMAX NTAB
VENTS	14808	VENTS.INS	ZZVENT VSS VSA VAS VAA VSAS VASA IZVENT NVENTS IVVENT NVVENT
VNTSLB	208	VNTSLB.INS	YVELEV DPV1M2 DIRS12 NVELEV IOUTF
WNODES	12324	WNODES.INS	WLENGTH WALLDX WSPLIT NUMNODE NWALLS IWBOUND NWPTS

Appendix E: Format of HV1.CF and INITSLV

INITSLV BLOCK DATA

These are all free format:

```
C     ABS PRESSURE TOL, REL PRESSURE TOL, ABS OTHER TOL, REL OTHER TOL
      DATA APTOL, RPTOL, ATOL, RTOL/1.D-5, 1.D-5, 1.D-2, 1.D-2/
C     ABS WALL TOL, REL WALL TOL
      DATA AWTOL, RWTOL, ALGTOL/1.4D-2, 1.4D-2, 1.D-8/
C     ABS HVAC PRESS, REL HVAC PRESS, ABS HVAC TEMP, REL HVAC TEMP
      DATA AHVPTOL,RHVPTOL,AHVTTOL,RHVTTOL/1.D-5,1.D-5,1.D-2,1.D-2/
      DATA (OPTION(J),J=1,16)/
C     OPTIONS FIRE, HFLOW, ENTRAIN, VFLOW, CJET, DOOR-FIRE, CONVEC, RAD,
     +         2,    1,      1,       1,     1,      1,        1,    1,
C         CONDUCT, DEBUG, EXACT ODE,  HCL , MFLOW, KEYBOARD,
     +         1,    0,     1,        1,     1,      0,
C         TYPE OF INITIALIZATION,   MV HEAT LOSS
     +         1,                      0/
C     NUMBER OF WALL NODES, FRACTIONS FOR FIRST, MIDDLE AND LAST WALL SLAB
      DATA NWPTS /20/
C     BOUNDARY CONDITION TYPE (1=CONSTANT TEMPERATURE, 2=INSULATED 3=FLUX)
      DATA IWBOUND /1/
C     COMPUTED VALUES FOR BOUNDARY THICKNESS
      DATA (WSPLIT(J),J=1,3)  /0.65, 0.15, 0.20/
C     TURN DEBUGGING OPTIONS OFF - THIS IS NOT CURRENTLY USED
      DATA DEBUG /MXOPT*0/
C     MAXIMUM STEP SIZE***, MAXIMUM FIRST STEP: IF NEGATIVE THEN SOLVER WILL DECIDE
      DATA STPMAX /-1.0/, DASSLFTS/-1.0/
```

HV1.CF

header (a6): $$CF$$
version, color table: 1960 15 14 15 11 15 15 0 1 1 0 12 6 2 10 13 4 16 16 1T
file name for the thermophysical properties (a60): THERMAL.DF (default is THERMAL.DAT)
restart file (a60): BURN.DAT
geometric database (a60): GEOMETRY.DAT
multiple objects database (a60): OBJECTS.DF (default is OBJECTS.DAT)
partitian data base (a60): PARTITIN.DAT
data base path (a64): \MODELS\CFAST1P6\BIN (defaults to current directory if blank)
data directory (a64): \MODELS\CFAST1P6\BIN\DATA (defaults to current directory if blank)
default graphics display (for pc displays) (a2,i1): IV3

*** This value may be modified by placing the keyword STPMAX into the CFAST data file followed by the desired value for the maximum step size (for example STPMAX 3.0 will set the maximum step size for the solver to 3 s). Normally, it may be left at the default value to allow the solver to chose the step size automatically.